Titre: Title:	Architecture et mécanismes pour la gestion de la qualité de service dans les réseaux de prochaines générations
Auteur: Author:	Khyda Désiré Yannick Oulaï
Date:	2007
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Oulaï, K. D. Y. (2007). Architecture et mécanismes pour la gestion de la qualité de service dans les réseaux de prochaines générations [Ph.D. thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/8063/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/8063/
Directeurs de recherche: Advisors:	
Programme: Program:	Unspecified

NOTE TO USERS

UNIVERSITÉ DE MONTRÉAL

ARCHITECTURE ET MÉCANISMES POUR LA GESTION DE LA QUALITÉ DE SERVICE DANS LES RÉSEAUX DE PROCHAINES GÉNÉRATIONS

KHYDA DÉSIRÉ YANNICK OULAÏ
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIAE DOCTOR
(GÉNIE INFORMATIQUE)
MAI 2007

Your file Votre référence
ISBN: 978-0-494-35517-6
Our file Notre référence
ISBN: 978-0-494-35517-6

UNIVERSITÉ DE MONTRÉAL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

ARCHITECTURE ET MÉCANISMES POUR LA GESTION DE LA QUALITÉ DE SERVICE DANS LES RÉSEAUX DE PROCHAINES GÉNÉRATIONS

présentée par : OULAÏ Khyda Désiré Yannick

en vue de l'obtention du diplôme de : Philosophiae Doctor

a été dûment acceptée par le jury d'examen constitué de :

M. PESANT Gilles, Ph.D., président

M. PIERRE Samuel, Ph.D., membre et directeur de recherche

M. CHAMBERLAND Steven, Ph.D., membre et codirecteur de recherche

M. QUINTERO Alejandro, Doct., membre

M. WESSAM Ajib, Ph.D., membre

À ma femme, Laurette

REMERCIEMENTS

Je veux tout d'abord bénir le nom de mon Seigneur Jésus-Christ, source intarissable de réconfort et d'inspiration sans qui je ne serais pas ce que je suis.

J'aimerais aussi remercier MM Samuel Pierre et Steven Chamberland, directeur et codirecteur de recherche dont l'encadrement et les conseils m'ont guidé tout au long de ma recherche. Je tiens également à mentionner la collaboration essentielle du personnel de Ericsson Canada, notamment MM Laurent Marchand et Yves Lemieux.

Papa et maman, je vous embrasse et du fond du cœur je vous remercie d'avoir été présents par l'éducation, la confiance et l'appui moral et financier inconditionnel que vous m'avez apporté. À mes frères Jean-Marie, Stéphane et Grégory, mes oncles Germain et Marcellin ainsi que leurs tendres moitiés, à toute ma famille, je dis merci pour ces moments où vous m'avez fait ressentir la chaleur fraternelle.

À tous ceux qui m'ont gardé dans leurs prières, l'EEPH, les groupes Ministères et Autorités, les pasteurs Eugène et Célestin, Yves Zady, maman Lucie et le groupe des « mamies », Harry, Béatrice, Marie Josée, Jean-Charles, Patricia, Yao, Diane, Yves Trazo, Vanessa et bien d'autres, je vous remercie de ce temps précieux investi pour moi.

Mika, il n'y a pas deux sœurs comme toi! Ton écoute, tes conseils et tes prières m'ont permis de me relever dans des périodes difficiles.

À Marc, Raouf, Ali, Christian, Paul, Stéphane et à tous mes collègues du LARIM, je dis merci pour l'ambiance chaleureuse et enrichissante du laboratoire. Merci à Christiane et Linda pour la relecture de cette thèse. Je remercie aussi tous ceux que je n'ai pas eu la possibilité de nommer et qui m'ont soutenu, aidé ou encouragé d'une manière ou d'une autre durant ce long processus.

Enfin, je veux terminer en remerciant spécialement celle qui partage ma vie et à qui cette thèse est dédicacée, source de bonheur et de joie, ma Lau, dont l'affection et la présence ont été déterminantes tout au cours de cette recherche. Merci pour tout.

RÉSUMÉ

Durant la décennie 1990, le développement des réseaux de télécommunications a été fulgurant. De la téléphonie fixe à la téléphonie mobile, des données de second ordre aux données sécurisées, tous ces domaines ont connu d'importantes améliorations ainsi qu'une croissance vertigineuse du trafic. Une multitude de services comme la téléphonie mobile, les courriels, la vidéoconférence, la vidéo sur Internet ou le divertissement en ligne sont disponibles et requièrent des niveaux de plus en plus élevés de qualité de service. Toutes ces nouvelles technologies ont fait ressortir la nécessité de définir une architecture globale de réseaux de prochaines générations. Cette architecture permettra une convergence des réseaux fixes et mobiles. Ces réseaux devront offrir une vaste gamme de services ayant des requis de qualité de service variés sur une architecture commune utilisant le protocole IP. Plusieurs regroupements comme le *Multiservice Switching Forum* et *Telecoms and Internet converged Services and Protocols for Advanced Networks* (TISPAN) travaillent d'arrache pied pour définir une architecture permettant d'atteindre ces objectifs mais certaines sections de ces architectures ne sont pas encore correctement définies.

Un aspect fondamental de cette nouvelle génération de réseaux réside dans la qualité de service offerte aux usagers. La qualité de service est un ensemble de requis qu'un système de télécommunications peut offrir lors d'une session. Les indices de performance que l'on retrouve très souvent sont le délai, la gigue, le débit, le taux de pertes de paquets, le taux d'erreurs, la robustesse et le taux de blocage.

À cause de l'évolution des services offerts, la qualité de service devient une notion incontournable pour les opérateurs. Premièrement, dans les périodes de congestion du réseau, il est primordial d'avoir des mécanismes pour un traitement différencié des données en circulation. Ensuite, face à la concurrence, un opérateur qui n'est pas en mesure de garantir la qualité de l'expérience des utilisateurs verra ses abonnés se tourner vers ses compétiteurs. Dans cette optique, nous avons choisi comme objectif principal de

cette thèse de proposer une architecture de qualité de service pour un réseau d'accès de prochaine génération. Cette architecture permet de garantir certains paramètres comme le délai et la perte de paquets de bout en bout. Par ailleurs, nous visons aussi à intégrer Ethernet dans le réseau d'accès afin de favoriser une réduction des coûts. Dans cette thèse, nous définissons l'architecture physique et fonctionnelle ainsi que des protocoles permettant un fonctionnement adéquat de nos propositions.

Pour atteindre ces objectifs, nous avons d'abord effectué une revue de littérature afin de connaître l'état de l'art dans les différents domaines étudiés. Pour nos travaux, nous avons considéré l'architecture TISPAN car elle semble être la plus complète. Le premier aspect que nous avons étudié concerne la définition du réseau d'accès de niveau 2 de la couche OSI qui conduit à une architecture globale du réseau TISPAN. Nous avons utilisé Ethernet comme technologie de réseau d'accès car c'est une technologie répandue qui favorise une réduction des coûts. Cela nous a conduit à définir une architecture centralisée de gestion de qualité de service pour les réseaux Ethernet. Nous avons ensuite présenté une architecture pour le réseau d'accès TISPAN en intégrant l'architecture Ethernet que nous avions proposée afin de garantir les paramètres de qualité de service de bout en bout.

Suite à cela, nous avons développé des algorithmes et des modèles mathématiques pour faire le contrôle d'admission de connexions en temps réel dans un réseau avec une topologie logique prédéfinie comme MPLS. Préalablement, nous avions étudié les différents modèles de contrôle d'admission disponible dans la littérature afin d'en déceler les forces et les faiblesses.

La dernière phase de ce travail a consisté à évaluer nos propositions. Les protocoles de réservation de ressources dans le réseau d'accès TISPAN ont été validés formellement avec le logiciel UPPAAL. Pour le contrôle d'admission, la génération des modèles se fait à partir d'un code en langage C tandis que les modèles sont résolus avec le logiciel CPLEX. Nos résultats ont prouvé que nos propositions permettent de satisfaire toutes les connexions en service sans augmenter significativement le blocage. Étant donné que nous sommes dans un cadre de contrôle d'admission dynamique, nous avons démontré que nos

modèles se résolvent en moins de 53 ms avec le logiciel CPLEX, avec un temps d'exécution total en dessous de 260 ms et ce, pour des réseaux de grande taille. Au regard de ces résultats, nous pouvons affirmer que nos propositions sont intéressantes et pourraient être appliquées dans de vrais réseaux.

Cette thèse apporte des contributions majeures et originales au domaine des réseaux de télécommunications. Le premier apport significatif est la proposition d'une architecture de qualité de service pour les réseaux Ethernet. En effet, Ethernet n'a pas été conçu à la base pour supporter des applications avec des requis stricts de qualité de service. Nous avons donc proposé une architecture centralisée qui permet de faire le contrôle d'admission de connexions, l'application des politiques et le marquage des trames. Le contrôle d'admission est effectué par un nœud central tandis que les nœuds frontières se chargent de l'application des politiques et du marquage des trames. Cette architecture permet de garantir des paramètres de qualité de service à des flots individuels tout en conservant une approche *DiffServ*. Le fonctionnement des commutateurs internes au domaine Ethernet s'en trouve simplifié car ils ne conservent pas d'états sur les différents flots.

Pour permettre une meilleure différentiation des classes de service dans un réseau Ethernet, nous avons mis au point une solution permettant d'augmenter le nombre de niveaux de priorités qui est limité à 8 avec la norme IEEE 802.1Q. Cette solution permet aussi de définir de nouveaux VLAN ou de transporter des étiquettes pour la commutation des trames Ethernet. Un des avantages principaux de cette proposition est que la taille de l'entête Ethernet est conservée, ce qui permet une compatibilité avec les équipements actuels.

En outre, l'intégration d'Ethernet au réseau TISPAN est originale car aucune proposition n'a été faite dans ce sens. Nous avons aussi défini des protocoles de réservations de ressources pour le réseau d'accès TISPAN.

Par ailleurs, les modèles de programmation mathématique pour modéliser un contrôle d'admission de connexion en temps réel permettent de résoudre trois types de problèmes avec contraintes de délai, contraintes de perte de paquets et contraintes mixtes.

Toutes les contraintes considérées sont de bout en bout. Sur ce point, plusieurs aspects d'originalité sont à noter. Premièrement, les modèles proposés permettent de garantir les paramètres de qualité de service de tous les flots en service sur le réseau sans rerouter les connexions. En effet, contrairement à la majorité des propositions de contrôle d'admission qui se focalisent sur la satisfaction des paramètres de qualité de service de bout en bout pour la connexion qui demande l'accès au réseau sans tenir compte des trafics déjà en service, nous assurons que, pour toutes les connexions, les paramètres de qualité de service sont maintenus en deçà des seuils qui ont été négociés avec l'opérateur de réseau. Ce faisant, l'ajout d'une nouvelle connexion sur le réseau ne dégrade pas l'expérience des autres utilisateurs, ce qui permettra à l'opérateur de satisfaire ses clients et donc de les conserver. Ensuite, les modèles proposés sont linéaires, ce qui permet une résolution rapide avec des logiciels comme CPLEX. Enfin, les algorithmes utilisés favorisent la réduction du nombre de contraintes. En effet, au lieu d'écrire une contrainte pour chaque connexion en service sur le réseau, nous avons adopté une approche par chemin. Elle consiste à n'écrire une contrainte pour un chemin donné que si au moins une connexion est susceptible de dépasser les seuils de qualité de service négociés en cas d'acceptation de la nouvelle connexion sur au moins un lien du chemin considéré. Cette approche favorise un temps de réponse rapide lors d'une requête de connexion.

ABSTRACT

During the nineties, telecommunications networks evolved significantly in matters of technology and quantity of traffic. Many services like voice, video, videoconference and online gaming require higher and higher level of quality of service. All these new technologies point out the need to define a global architecture for next generation networks. Next generation networks are IP-based networks and will be able to support both fixed and mobile applications with various levels of quality of service. Many standardizations bodies like *Multiservice Switching Forum* et *Telecoms and Internet converged Services and Protocols for Advanced Networks* (TISPAN) work to define next generation networks architectures but there are still some unresolved issues.

A fundamental aspect for these future networks is quality of service for users. Quality of service is a set of requirements that a network shall provide. Common performance indicators are delay, jitter, throughput, packet loss rate and error rate. Due to the real time nature of new applications, an operator unable to guarantee a certain level of quality of service may loose some of his subscribers. That is why the main objective of this thesis is to propose a quality of service architecture for a next generation access network. This architecture allows guaranteeing some end to end parameters like delay and packet loss rate.

To reach our goals, we first reviewed the literature in order to evaluate the state of the art in the different area we had to study. This thesis considers the TISPAN network which is the most complete at this time. The first aspect we studied is the definition of a layer 2 aggregation network which leads to a global architecture. We used *Ethernet* as access technology because *Optical Ethernet* has a low cost/performance ratio with high bitrate available. We started by defining a centralized architecture for quality of service management in Ethernet networks. Then, in order to guarantee the end to end quality of service, we presented an architecture for the TISPAN access network which integrates the Ethernet architecture previously defined.

Following that, we proposed some algorithms and mathematical models for real time connection admission control in a network with a predefined logical topology. Beforehand, we analyzed the different models of admission control already proposed in the literature in order to detect their forces and weaknesses. We focused on end to end delays and packet loss constraints.

The last part of our work was the evaluation of our solutions. The protocols for resource reservation in a TISPAN access network were formally validated using a software named UPPAAL. For admission control, models are generated using a C code and solved with CPLEX software. Our results proved that our proposals permit to satisfy every connexion in service without significantly increase the blocking. Since we are in a dynamic context, we showed that it takes under 53 ms to solve our models with CPLEX and the total running time is less than 260 ms for 80 nodes networks. Our proposals could therefore be applied to real networks.

This thesis brings some major and innovative contributions to the telecommunication networks area. First, the quality of service architecture for Ethernet is important because Ethernet was not originally designed for real time services. Therefore, we proposed a centralized architecture that allows doing connections admission control, policy enforcement and frames marking. This solution is *DiffServ*-based but allows guaranteeing some quality of service parameters to individual flows.

For a better class of service differentiation in an Ethernet network, we introduced a mechanism to increase the number of priority level which is limited to eight with the IEEE 802.1Q standard. Moreover, it is also possible to increase the number of VLAN or to carry label for Ethernet label switching. One of the main advantages of this proposal is that the Ethernet frame header length remains the same, allowing backward compatibility with existing equipments. Also, the integration of Ethernet in the TISPAN network is innovative.

On the other hand, our mathematical programming models for admission control permit to solve problems with three kinds of constraints: delays, packet loss and mixed constraints. All the constraints are end to end. For this part of our work, we should

mention many new features. First, the proposed models allow guaranteeing quality of service parameters for every connexion in the network without rerouting while most of the propositions in the literature only focus on guaranteeing quality of service for the new connection. With this approach, accepting a new connection on the network will not perturb the connections already in service and the operator will retain his subscribers. Another point is that the proposed models are linear, leading to fast resolutions. Finally, the proposed algorithms favour the reduction of the number of constraints to write because we adopt a path-based approach rather than a flow based approach.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

3GPP	Third Generation Partnership Project
AN	Access Node
A-RACF	Access- Resource and Admission Control Function
ATIS	Alliance for Telecommunications Industry Solutions
BGF	Border Gateway Function
BM	Bandwidth Manager
B-TAG	Backbone TAG
B-VLAN	Backbone VLAN
CAC	Contrôle d'admission
CACM	CAC basée sur les mesures
CACR	CAC basée sur les réservations
CDMA2000	Code Division Multiplexing 2000
CdS	Classe de Service
CMS	Call Management Server
CMTS	Cable Modem Termination System
CPE	Customer Premise Edge
C-TAG	Client TAG
C-VLAN	Client VLAN
DHCP	Dynamic Host Control Protocol
DiffServ	Differentiated Services
DMQ	Délai de mise à jour de QdS
DNS	Domain Name Server
DOCSIS	Data Over Cable Service Interface Specification
ELS	Ethernet Label Switching
EMS	Element Management System
E-MTA	Embedded MTA
EoMPLS	Ethernet over MPLS
ETSI	European Telecommunications Standards Institute
EVC	Ethernet Virtual Circuit
IEEE	Institute of Electrical and Electronics Engineers
IMS	IP Multimedia Subsystem
IntServ	Integrated Services
I-SID	service Instance TAG
I-TAG	service Instance ID
ITU-T	International Telecommunication Union - Telecommunication

KDC	Key Distribution Protocol
L2LSP	Layer 2 LSP
MEF	Metro Ethernet Forum
MetroEthernet	Metropolitain Ethernet
MG	Media Gateway
MGC	Media Gateway controller
MSF	Multiservice Switching Forum
MTA	Multimedia Terminal Adapter
N_D	Nœud Destination
N_F	Nœud Frontière
N_{FE}	Noeud frontière d'entrée
N_{FS}	Noeud frontière de sortie
N_I	Nœud Interne
N_S	Nœud Source
NE	Network Element
NMS	Network Management System
NSIS	Next Step In Signalling
OSS	Operation System Support
OUI	Organization Unique Identifier
PHB	Per Hop Behaviour
PSTN	Public Switched Telephony Network
PW	Pseudo Wire
RACS	Resource and Admission Control Subsystem
RCEF	Resource Control Enforcement Point
RSVP	Resource reSeVvation Protocol
SBG	Session Border Gateway
SBM	Subnet Bandwidth Manager
SG	Signalling Gateway
SIP	Session Initiation Protocol
SONET	Synchronous Optical NETwork
ST	Spanning Tree
S-TAG	Service TAG
S-VID	Service VID
S-VLAN	Service VLAN
TCI	Tag Control Identifier
TG	Trunking Gateway
TISPAN	Telecoms & Internet converged Services & Protocols for Advanced Networks

TPID	Tag Protocol Identifier
UMTS	Universal Mobile Telecommunications System
VID	VLAN Identifier
VLAN	Virtual Local Area Network
VPLS	Virtual Private Network
VPN	Virtual Private Network

LISTE DES ANNEXES

CHAPITRE I

INTRODUCTION

La fin du 20^{e} siècle a connu le développement fulgurant des réseaux de télécommunications. Une multitude de services comme la téléphonie mobile, les courriels, la vidéoconférence, la vidéo sur Internet ou le divertissement en ligne sont disponibles et requièrent des niveaux de plus en plus élevés de qualité de service (QdS). Toutes ces nouvelles technologies ont fait ressortir la nécessité de définir une architecture globale de réseaux de prochaines générations (*Next Generation Network* : NGN). Les NGN seront basés sur le protocole IP qui s'impose comme protocole de la couche réseau aussi bien dans les réseaux mobiles que fixes. Les NGN permettront aussi une convergence des réseaux fixes et mobiles. Par ailleurs, les NGN visent à favoriser l'utilisation de technologies ayant un rapport coût/performance intéressant. On peut citer comme technologie candidate *Optical Ethernet* pour le faible rapport coût/bande passante impliquée par l'utilisation de commutateurs Ethernet optiques.

Toutefois, malgré la maturité de ces technologies, l'aspect de qualité de service n'est pas encore totalement maîtrisé. C'est pourquoi cette thèse se propose d'étudier les réseaux Ethernet et les réseaux IP par rapport aux aspects de QdS dans les NGN. Le but ultime est de proposer des mécanismes pouvant s'intégrer à une architecture globale de NGN. La première partie de la thèse consiste à définir une architecture de QdS pour les réseaux Ethernet métropolitains. Cette architecture devra tenir compte des évolutions actuelles d'Ethernet. Elle sera ensuite intégrée au réseau d'accès du NGN. Puis, nous proposerons des heuristiques permettant de faire le contrôle d'admission de connexion en tenant compte des critères de QdS.

Dans ce chapitre, après avoir introduit les réseaux de prochaines générations, nous ferons ressortir les éléments de la problématique qui seront suivis par un énoncé des objectifs de recherche que nous avons déterminés. L'esquisse méthodologique précédera

l'énoncé des principales contributions de la thèse. Enfin, le plan de la thèse sera le dernier point de ce chapitre.

1.1 Réseaux de prochaines générations

De nos jours, la nécessité est apparue de définir des architectures de réseaux de prochaines générations basées sur les paquets et permettant de supporter un vaste ensemble de services multimédia ainsi que la téléphonie classique, tout en offrant la qualité de service nécessaire. Dans les NGN, les fonctions de service sont indépendantes des fonctions de transport. Ces réseaux supportent différents fournisseurs de services. Par ailleurs, un des objectifs fondamentaux est de fournir une QdS de bout en bout. Les NGN peuvent être interconnectés à des réseaux classiques tout en respectant certaines contraintes légales et administratives. Il faut aussi mentionner qu'un des objectifs majeurs des NGN est la convergence des réseaux fixes et mobiles vers des réseaux unifiés en utilisant le protocole IP.

Plusieurs organismes s'attèlent à la définition de ce type d'architecture. L'*International Telecommunication Union - Telecommunication* (ITU-T) est l'organisme international chargé de la standardisation globale de l'architecture des NGN. À cet effet, cet organisme a produit un ensemble de requis génériques dont s'inspirent certaines structures régionales (ITU-T, 2004a, ITU-T, 2004b). Parmi les organismes régionaux de normalisation, on retrouve en Europe le groupe *Telecoms and Internet converged Services and Protocols for Advanced Networks* (TISPAN) de l'*European Telecommunications Standards Institute* (ETSI) qui a défini le TISPAN NGN basé sur le *IP Multimedia Subsystem* (IMS) de l'organisme *Third Generation Partnership Project* (3GPP) qui a été adapté pour les réseaux fixes (TISPAN, 2005a). L'architecture de TISPAN NGN est indépendante du type de réseau d'accès utilisé. Aux Etats-Unis, *Alliance for Telecommunications Industry Solutions* (ATIS) essaie de développer une architecture de NGN qui puisse répondre aux besoins du marché nord américain tout en s'insérant dans le cadre global de l'ITU-T (ATIS, 2004). Toutefois, la définition de l'architecture NGN de ATIS est à ses tous débuts et s'inspire grandement des travaux de

TISPAN. Par ailleurs, le *Multiservice Switching Forum* (MSF) est un forum qui vise aussi à développer une architecture ouverte multiservices pouvant fonctionner avec un large éventail de technologies et des systèmes conçus par différents opérateurs (MSF, 2005b). L'objectif principal de ce regroupement est d'accélérer la disponibilité commerciale et l'interopérabilité des systèmes. Pour ce faire, des démonstrations d'interopérabilité à large échelle sont réalisées. MSF favorise aussi l'innovation et le déploiement rapide et peu coûteux de services à valeur ajoutée.

D'autres organismes de standardisation comme *PacketCable,* 3GPP et 3GPP2 interviennent aussi dans la définition des NGN à cause de leur expérience dans certains domaines particuliers. *PacketCable* est une initiative de *CableLabs*, un regroupement d'opérateurs œuvrant dans le domaine de la télévision par câble (PacketCable, 2005a). Au départ, leur intention était de profiter de la popularité du câble pour offrir de la voix sur IP mais différents types de services multimédia sont aujourd'hui offerts. Le réseau *PacketCable* est bâti sur le réseau hybride de fibres et câbles coaxiaux des opérateurs utilisant DOCSIS (PacketCable, 2005b) qui est une norme spécifique permettant de transporter des données à haute vitesse à partir des modems câble utilisés pour la télévision.

Par ailleurs, 3GPP définit une architecture dont le réseau d'accès est basé sur l'UMTS. Dans sa version 5, elle inclut un IMS qui permet d'offrir des services multimédia de qualité (3GPP, 2005). Cet IMS, additionné de quelques extensions, représente la portion du réseau cœur des NGN capables de fournir des services multimédia. D'autre part, 3GPP2 est un groupe qui définit une architecture de services multimédia assez similaire à celle de 3GPP hormis que celle de 3GPP2 est conçue pour développer le CDMA2000 tandis que celle de 3GPP s'est basée sur l'UMTS (3GPP2, 2003).

Dans les NGN comme dans tout type de réseau, il y a souvent des portions de réseau basées sur des technologies de la couche 2 du modèle OSI comme *Asynchronous Transfer Mode* (ATM) et Ethernet. Plusieurs études font abstraction de la portion niveau 2 mais si la QdS de bout en bout est étudiée, il apparaît important de tenir compte de la globalité du réseau. Ethernet est une technologie de niveau 2 qui est très répandue dans

les réseaux locaux. Selon Chiruvolu *et al.* (2004), plus de 90% du trafic IP est issu de LAN Ethernet. Ethernet a été standardisé dans le début des années 80 avec les normes Ethernet II DIX et IEEE 802.3 (IEEE, 1996). La différence fondamentale entre ces deux normes réside dans le champ *Ethertype* qui, pour Ethernet 2 représente le protocole de couche supérieure et pour IEEE 802.3 la taille des données. Toutefois, la plupart des trames utilisées de nos jours contiennent l'Ethertype. L'entête de la trame Ethernet se compose d'un préambule de 8 octets permettant la synchronisation des équipements, des adresses MAC destination et source de 6 octets chacune et du champ Ethertype ou Longueur selon le type de trame. Les réseaux Ethernet fonctionnent en mode *broadcast* et certains commutateurs spécialisés permettent de séparer les domaines de collisions.

Ethernet, connu pour être une technologie de LAN, tend à s'implanter dans les réseaux de plus grande envergure grâce à la technologie *Optical Ethernet* qui propose des commutateurs optiques Ethernet à haut débit intégrant des fonctions de commutations Ethernet en plus de fonctions optiques. Les liens classiques, dont la portée était de quelques centaines de mètres, sont remplacés par des liens optiques pouvant s'étirer sur quelques kilomètres et offrant une largeur de bande supérieure. Ce faisant, cette technologie pourrait être une pièce maîtresse des NGN. En effet, en plus d'être peu onéreux, le développement de services basés sur Ethernet fait de cette technologie une excellente candidate pour les réseaux métropolitains et plusieurs opérateurs pensent à l'intégrer comme réseau d'accès (Chiruvolu et al., 2004). Le terme MetroEthernet est utilisé pour désigner l'Ethernet Métropolitain.

Un avantage fondamental d'Ethernet réside dans sa prépondérance dans les LAN, aussi bien dans les entreprises que chez les particuliers. Les entreprises ont adopté ce standard essentiellement à cause du rapport prix/performance intéressant et de leurs besoins grandissants en bande passante. En effet, Ethernet offre de nos jours une largeur de bande importante pouvant atteindre 10 gigabits par seconde (ATRICA, 2003, NSP, 2003). En plus, l'automatisation de la gestion des réseaux Ethernet permet d'attribuer dans un délai de quelques minutes la largeur de bande aux clients comparativement aux mois nécessaires pour les technologies actuelles. D'autre part, Ethernet se prête bien au trafic

en rafales, ce qui n'est pas le cas de *Synchronous Optical NETwork* (SONET) par exemple. En outre, le transport de trames Ethernet sur les réseaux *Multiprotocol Label Switching* (MPLS) est un aspect qui retient aussi l'attention des chercheurs qui voudraient conserver le format de la trame Ethernet de bout en bout. Enfin, même s'il est adressé, l'aspect QdS Ethernet pose un problème à certains égards que nous détaillerons au chapitre 2.

Un paramètre important de nos jours, et qui le sera encore plus lors du déploiement des NGN, est la QdS de bout en bout. Chaque application multimédia ou temps réel a des caractéristiques propres quant à certains indices de performance. La QdS est un ensemble de requis qu'un système de télécommunications peut offrir lors d'une session. Les indices de performance que l'on retrouve très souvent sont le délai, la gigue, le débit, le taux de pertes de paquets, le taux d'erreurs, la robustesse, le taux de blocage, le taux d'interruption des appels et le délai de mise à jour de la QdS.

Au niveau IP, il existe principalement deux types de QdS : *IntServ* et *DiffServ*. *IntServ* se base sur une architecture de QdS permettant de réserver les ressources en fonction des requis de chaque application et de la politique de gestion des ressources. On peut citer *Resource reSerVation Protocol* (RSVP) comme un protocole de signalisation pouvant supporter *IntServ*. *DiffServ* permet de gérer des agrégats de flots en séparant le trafic selon certaines classes prédéfinies et en tenant compte de la méthode de gestion de capacité sur les liens du réseaux. Ces deux architectures peuvent être utilisées de manière complémentaire. *IntServ* permet une meilleure garantie de la QdS mais peut rencontrer un problème au niveau de la gestion d'un grand nombre de flots. Toutefois, *IntServ* peut aussi être utilisé dans une approche basée sur les classes, ce qui pourrait quelquefois dégrader la QdS des flots individuels. *IntServ* s'adapte mieux aux réseaux d'accès dont le trafic est moins important que les réseaux dorsaux tandis que, *DiffServ*, approche plus évolutive est appropriée pour les réseaux dorsaux (Bernet, 2000).

Par ailleurs, RSVP est un protocole de signalisation permettant d'effectuer de la réservation de ressources pour des sessions *unicast* et *multicast* (Braden *et al.*, 1997). Il comprend un certain nombre de messages de signalisation servant à établir le chemin du

flot, réserver les ressources, rafraîchir et annuler les réservations. C'est un protocole orienté récepteur car c'est ce dernier qui émet le paquet chargé d'effectuer la réservation. Comme RSVP prend en compte des sessions *multicast*, cela en fait un protocole relativement complexe. RSVP est un utilisé dans plusieurs domaines, notamment dans l'optique pour faire la signalisation par rapport aux longueurs d'onde.

1.2 Éléments de problématique

Les NGN joueront un rôle important dans l'avenir des télécommunications à cause de la convergence entre les réseaux fixes et les réseaux mobiles et l'introduction de services à valeurs ajoutées. Le TISPAN NGN représente un candidat de choix par rapport à l'aspect architectural. Ce NGN semble complet et performant mais il comporte une lacune en ce sens que la technologie et l'architecture de la portion du réseau d'accès comprise entre le nœud d'accès au réseau et le premier routeur IP ne sont pas définies, ce qui peut influencer négativement la QdS. Pour assurer une QdS de bout en bout, il apparaît nécessaire de se pencher sur la question relative à cette portion de réseau tout en tenant compte de la gestion des ressources. Dans ce cadre, Ethernet semble être une solution intéressante pour les réseaux d'accès à cause de la réduction des coûts. Toutefois, pour adopter cette approche, il faut adresser adéquatement la question de la QdS quand on sait que beaucoup d'applications, et plus particulièrement les applications mobiles, ont des requis stricts de QdS. La plus fine granularité définie pour le MetroEthernet est la classe de service défini sur 3 bits. La concordance entre cette approche et les différents paradigmes de QdS IP en général mérite d'être approfondie. Il pourrait être intéressant d'augmenter le nombre de bits de priorité afin de définir plus de classes. En effet, un opérateur pourrait décider de facturer des services de voix de plusieurs manières en fixant des seuils différents pour les paramètres de QdS tels la disponibilité du réseau et le délai. On pourrait aussi imaginer que les trames d'un service de voix donné puissent être jetées prioritairement en cas de congestion. Par ailleurs, il peut être avantageux de créer une classe spéciale où se retrouveraient les appels d'urgence afin qu'ils ne rentrent en conflit avec aucun autre type de trafic. Aussi, puisque l'approche *IntServ* avec RSVP s'adapte

plus aux réseaux d'accès, la concordance entre les aspects de QdS de RSVP et d'Ethernet est importante afin de ne pas dégrader la QdS fournie par RSVP lors du passage dans l'environnement Ethernet.

En plus du modèle de QdS à définir, il se pose le problème de l'architecture fonctionnelle spécifique permettant de gérer le réseau Ethernet et d'interagir avec les autres éléments de QdS afin d'assurer la QdS globale dans le NGN. Les propositions faites pour des architectures de QdS Ethernet s'adaptent difficilement au contexte de réseaux métropolitains.

Une autre avenue de recherche intéressante consiste à étudier la commutation de trames Ethernet dans les environnements MPLS et GMPLS. Les chercheurs tendent à définir une étiquette de commutation à partir de l'entête Ethernet en limitant le plus possible la taille de l'entête. Dans ce cadre, certaines des idées avancées entraînent une modification fondamentale du fonctionnement de Ethernet, ce qui peut poser des problèmes d'interopérabilité lorsqu'une trame traverse un domaine constitué de commutateurs classiques.

D'autre part, quelles que soient l'architecture de QdS implantée et la technologie d'acheminement des trames (Ethernet natif ou MPLS), la question du contrôle d'admission des connexions (CAC) se pose. Le CAC est un processus qui, en fonction de l'état actuel du réseau, autorise ou rejette les connexions. C'est un aspect important de la gestion de la QdS car il permet de limiter l'accès au réseau afin de garantir certains critères de QdS. Ce faisant, les applications temps réel pourraient être correctement desservies dans les réseaux. Malheureusement, peu de travaux sur le CAC tiennent compte du délai et de la perte de paquets de bout en bout des connexions déjà établies sur le réseau. En effet, la plupart des processus de CAC proposés ne tiennent compte que des contraintes de QdS de bout en bout de la connexion qui demande l'accès au réseau sans s'assurer que les connexions en service respectent ces mêmes contraintes de chemins. En effet, tout usager espère conserver le niveau de QdS demandé pour toute la durée de la connexion. Une mauvaise gestion du CAC peut donc résulter en de mauvaises expériences pour les utilisateurs qui seront tentés de changer d'opérateur de réseau.

1.3 Objectifs de recherche

Le principal objectif de cette thèse est de proposer une architecture de QdS pour un réseau d'accès NGN. Pour ce faire, nous subdiviserons cet objectif en sous-objectifs qui sont :

1. analyser l'existant sur les NGN, MetroEthernet, la QdS au niveau IP et Ethernet afin d'identifier les forces et les faiblesses des solutions proposées ;
2. proposer une architecture de gestion de QdS ainsi que des mécanismes permettant de garantir les critères de QdS dans les environnements MetroEthernet ;
3. proposer une architecture de réseau d'accès NGN tenant compte des aspects de QdS ;
4. proposer des méthodes de contrôle d'admission de connexions permettant de préserver les niveaux de QdS pour toutes les connexions en service ;
5. implémenter, valider et évaluer les performances de nos solutions en les comparants aux meilleures solutions existantes dans la littérature.

1.4 Esquisse méthodologique

L'analyse de la littérature scientifique dans le domaine nous a amené à considérer une architecture TISPAN NGN qui semble être une des plus complètes et avancées parmi les différentes propositions. Pour atteindre l'objectif 1, nous effectuerons une revue de littérature afin de connaître l'état de l'art dans les différents domaines étudiés. Une activité de veille technologique nous permettra de nous tenir à jour par rapport aux avancées liées à nos axes de recherche. Pour atteindre l'objectif 2, nous proposerons une architecture qui devra être en mesure d'offrir un certain niveau de QdS aux flots individuels dans un réseau Ethernet dont le modèle de QdS est basé sur *DiffServ*. Nous mettrons aussi de l'avant un mécanisme permettant d'augmenter le nombre de classes disponibles pour le réseau Ethernet. Ce mécanisme pourra également servir à la définition d'étiquettes pour la commutation basée sur l'entête Ethernet. Le troisième objectif sera atteint en intégrant un réseau Ethernet dans un réseau d'accès TISPAN NGN. Il faudra

définir les interfaces ainsi que les protocoles permettant une interopérabilité et une complémentarité des parties IP et Ethernet du réseau d'accès.

L'atteinte du quatrième objectif, qui constitue une part importante de cette thèse, sera faite en proposant des algorithmes basés sur des modèles de programmation mathématique pour le contrôle d'admission. Nous allons considérer des critères de QdS de bout en bout qui sont le délai et le taux de perte de paquets. Pour simplifier la résolution, les modèles proposés devront être linéaires. Les contraintes de délai et de perte de paquets seront considérées d'abord séparément avant d'être intégrées ensemble. Nous utiliserons les modèles de files d'attentes M/M/1 et M/M/1/k pour modéliser les liens. Par ailleurs, le temps de résolution sera un facteur important puisque le CAC doit se faire en temps réel.

Nos solutions seront ensuite implémentées, validées formellement ou évaluer analytiquement par rapport à l'objectif 5. L'architecture de QdS pour Ethernet et celle du réseau d'accès TISPAN NGN proposées seront validées en terme de fonctionnement global. Des modèles de validation formelle seront implémentés afin d'évaluer certaines propriétés liées à la réservation des ressources. Finalement, nous prévoyons utiliser le logiciel CPLEX pour résoudre les modèles mathématiques proposés pour le contrôle d'admission.

1.5 Principales contributions et originalité

Cette thèse apporte des contributions majeures et originales pour le domaine des réseaux de télécommunications. Le premier apport significatif est la proposition d'une architecture de QdS pour les réseaux Ethernet. En effet, Ethernet n'a pas été conçu pour supporter des applications avec des requis stricts de QdS. C'est pourquoi nous avons proposé une architecture centralisée qui permet de faire le contrôle d'admission de connexions, l'application des politiques et le marquage des trames. Le contrôle d'admission est effectué par un nœud central tandis que les nœuds frontières se chargent de l'application des politiques et du marquage des trames. Cette architecture permet de garantir des paramètres de QdS à des flots individuels tout en conservant une approche

DiffServ. Le fonctionnement des commutateurs internes au domaine Ethernet s'en trouve simplifié car ils ne conservent pas d'états sur les différents flots.

Pour permettre une meilleure différentiation des classes de service dans un réseau Ethernet, nous avons mis au point une solution permettant d'augmenter le nombre de classes de services qui est limité à 8 avec la norme IEEE 802.1Q. Nous exploitons le champ *Ethertype* de l'entête Ethernet qui est une information statique lors du passage dans un réseau MetroEthernet. Cet espace est donc réutilisé pour définir de nouvelles classes de services. Par ailleurs, les bits du champ *Ethertype* peuvent aussi servir à définir de nouveaux VLAN ou à transporter des étiquettes pour la commutation des trames Ethernet. Un des avantages principaux de l'utilisation de l'*Ethertype* est que la taille de l'entête Ethernet est conservée, ce qui permet une compatibilité avec les équipements actuels.

Après avoir présenté des solutions pour la gestion de la QdS au niveau Ethernet, nous avons proposé d'intégrer un réseau Ethernet dans un réseau d'accès TISPAN. Cela nous permet de jouir de la réduction des coûts apportée par Ethernet et de garantir la QdS de bout en bout dans le réseau d'accès. Dans notre solution, nous utilisons un contrôleur pour chacun des domaines IP et Ethernet mais la décision finale revient au contrôleur du domaine IP car celui-ci a une vue plus globale du réseau d'accès. Nous avons aussi défini des protocoles de réservations de ressources qui impliquent les différents contrôleurs.

En plus de la définition des architectures, nous avons abordé le sujet spécifique du contrôle d'admission des connexions dans un réseau MPLS. Nous avons défini des modèles de programmation mathématique pour le contrôle d'admission de connexions en temps réel. Nos modèles permettent de résoudre trois types de problèmes, à savoir :

1. le contrôle d'admission avec des contraintes de délai ;
2. le contrôle d'admission avec des contraintes de perte de paquets ;
3. le contrôle d'admission avec des contraintes mixtes de délai et de perte de paquets.

La difficulté de ces problèmes réside aussi dans le fait que nous considérons des contraintes de bout en bout.

Par rapport à nos modèles et aux algorithmes qui permettent de les bâtir, plusieurs aspects d'originalité sont à souligner :

1. les modèles proposés tiennent compte des paramètres de QdS de toutes les connexions en service sur le réseau. En effet, contrairement à la majorité des propositions de contrôle d'admission qui se focalisent sur la connexion qui demande l'accès au réseau sans tenir compte du délai et de la perte de paquets de bout en bout des trafics déjà en service, nous assurons que les paramètres de QdS sont maintenus en dessous des seuils qui ont été négociés avec l'opérateur de réseau. Ce faisant, l'ajout d'une nouvelle connexion sur le réseau ne dégrade pas l'expérience des autres utilisateurs, ce qui permettra à l'opérateur de conserver sa clientèle ;
2. les modèles proposés sont linéaires, ce qui permet une résolution rapide avec des logiciels comme CPLEX. La contribution en termes de linéarisation des modèles est plus marquante dans le cas des contraintes de perte de paquets car celles-ci sont multiplicatives ;
3. les algorithmes proposés permettent de réduire le nombre de contraintes. En effet, au lieu d'écrire une contrainte pour chaque connexion en service sur le réseau, nous avons adopté une approche par chemin. Elle consiste à n'écrire une contrainte pour un chemin donné que si au moins une connexion est susceptible de dépasser les seuils de QdS négociés en cas d'acceptation de la nouvelle connexion sur au moins un lien du chemin considéré. Cette approche favorise un temps de réponse rapide lors d'une requête de connexion.

1.6 Plan de la thèse

Cette thèse est répartie sur six chapitres. Le chapitre suivant présente les NGN, MetroEthernet, le contrôle d'admission et les différents mécanismes de QdS IP. Puis, dans le chapitre 3, nous présentons les solutions que nous proposons pour les réseaux Ethernet et TISPAN. Le chapitre 4 traite de notre proposition concernant le contrôle d'admission de connexion avec contraintes de délai. Le chapitre 5, quant à lui, expose

notre heuristique pour le CAC avec des contraintes de pertes de paquets. On y retrouve aussi l'algorithme de résolution d'un problème multi-contraintes. Mentionnons que les chapitres 3, 4 et 5 contiennent au besoin des sections sur la validation formelle et les simulations. Enfin, le chapitre 6 résume les travaux accomplis assortis des limitations et d'une esquisse des travaux futurs suggérés.

CHAPITRE II

TECHNOLOGIES ET ARCHITECTURES DES RÉSEAUX DE PROCHAINES GÉNÉRATIONS

Les NGN constituent la prochaine étape de l'évolution des réseaux de télécommunications. Plusieurs architectures sont actuellement à l'étude dans ce cadre. Les NGN permettront de faire cohabiter différentes technologies afin de fournir un large éventail de services aux usagers. Ce chapitre analyse les technologies et architectures des réseaux de prochaines générations. Nous présentons d'abord les requis et les architectures pour les NGN définis par différents organismes de standardisation. Ensuite, nous survolerons la technologie Ethernet pour les réseaux métropolitains, Ethernet permettant de répondre aux requis des NGN à cause de son rapport coût/performance intéressant. Pour finir, nous parlerons du contrôle d'admission de connexions, un aspect important de la gestion de QdS.

2.1 Requis et architectures des NGN

Cette section présente les requis fondamentaux des NGN ainsi que certaines propositions d'architectures. La plupart des travaux concernant la définition d'architectures pour les NGN ont été effectués par des organismes internationaux de standardisation.

2.1.1 ITU-T

L'ITU a surtout travaillé à la production de standards définissant les requis généraux des NGN (ITU-T, 2004a, ITU-T, 2004b). Dans les NGN, les services et le transport sont séparés. On retrouve la strate des services et la strate de transport. La strate des services est définie par la partie du réseau NGN assurant les fonctions de transfert de données

liées au service ainsi que les fonctions de commande et de gestion permettant d'assurer les services aux utilisateurs. La strate de transport représente la partie du réseau NGN assurant les fonctions de transfert de données, de commande et de gestion des ressources de transport. Voici quelques exemples de caractéristiques fondamentales proposées par l'ITU :

- réseau à commutation de paquets comme les réseaux IP ;
- large gamme de services et applications disponibles ;
- largeur de bande importante avec QdS de bout en bout ;
- convergence des services fixes et mobiles ;
- interconnexion avec les réseaux classiques ;
- technologies d'accès multiples ;
- mobilité généralisée qui réfère à la possibilité pour un usager ou un mobile de communiquer et d'accéder aux services quelque soit la localisation ou la technologie disponible.

Les organismes régionaux de standardisation s'inspirent généralement des requis formulés par l'ITU. Pour l'instant, l'aspect mobilité n'a pas encore été vraiment adressé. La plupart des architectures spécifiques sont développées dans les organismes régionaux de standardisation comme TISPAN en Europe et MSF en Amérique du Nord. Bien évidemment, l'atteinte de ces objectifs ne doit pas occulter l'aspect économique lié à la réalisation des NGN. D'un côté, il est important pour les usagers de réduire les coûts tout en bénéficiant de services évolués et de l'autre, les opérateurs veulent réduire les coûts d'opération tout en diversifiant les services offerts.

2.1.2 TISPAN

TISPAN définit une architecture comprenant des équipements et réseaux usagers, des réseaux d'accès et un réseau cœur (TISPAN, 2005a). Les équipements usagers sont de différents types et sont généralement reliés au réseau d'accès par des passerelles, des modems ou d'autres types d'équipements. Le réseau cœur est composé de différents sous systèmes dont le sous système IP Multimedia (IMS *Core*) inspiré du IMS de 3GPP. La

Figure 2.1 présente l'architecture globale de TISPAN. Le réseau cœur contient un IMS qui sert à fournir des services multimédia basés sur le protocole de signalisation SIP. Le PES (*PSTN/ISDN Emulation Subsystem*) permet l'émulation de services PSTN/ISDN pour les terminaux classiques de téléphonie raccordés au NGN. Les concepteurs prévoient aussi de mettre en place d'autres sous systèmes comme le sous système de diffusion qui servira à distribuer les contenus multimédia à plusieurs usagers simultanément. Par ailleurs, le réseau d'accès comporte une portion de technologie niveau 2 OSI qui n'a pas été définie ainsi qu'une portion IP.

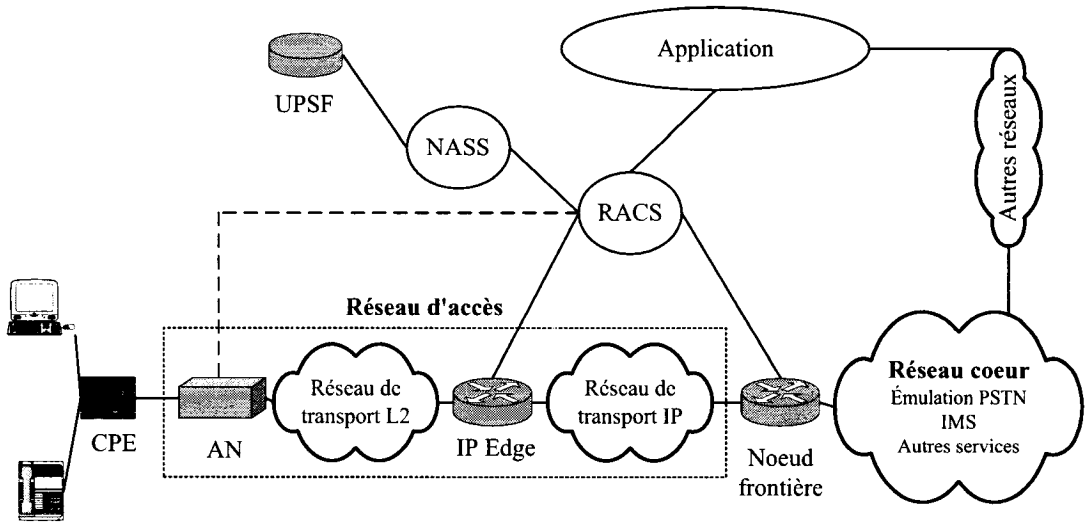

Figure 2.1 Architecture d'un réseau TISPAN NGN

Les profils des usagers sont conservés dans une base de données appelée UPSF (*User Profile Server Function*). Le NASS (*Network Attachment Subsystem*) permet d'attribuer des adresses et de gérer tout ce qui a trait à l'autorisation et à l'authentification des usagers en se basant sur le UPSF.

Il faut souligner que la QdS de bout en bout est un aspect important des NGN. C'est pourquoi le contrôle d'accès est effectué par le RACS (*Resource Admission Control Subsystem*). Ce dernier est en charge d'autoriser ou non une session en fonction du profil de l'usager, des politiques en vigueur, de la quantité maximale de ressources pouvant lui

être allouée ainsi que de la disponibilité de ces dernières. Le RACS se charge aussi de la réservation de ressources et supporte des modèles de QdS garanties ou relatives comme *DiffServ* (TISPAN, 2005b). La Figure 2.2 donne une vue fonctionnelle du RACS. L'équipement qui fait l'interface entre le réseau et l'usager (CPE: *Customer Premise Edge*) est connecté au réseau d'accès par un nœud d'accès (AN: *Acces Node*).

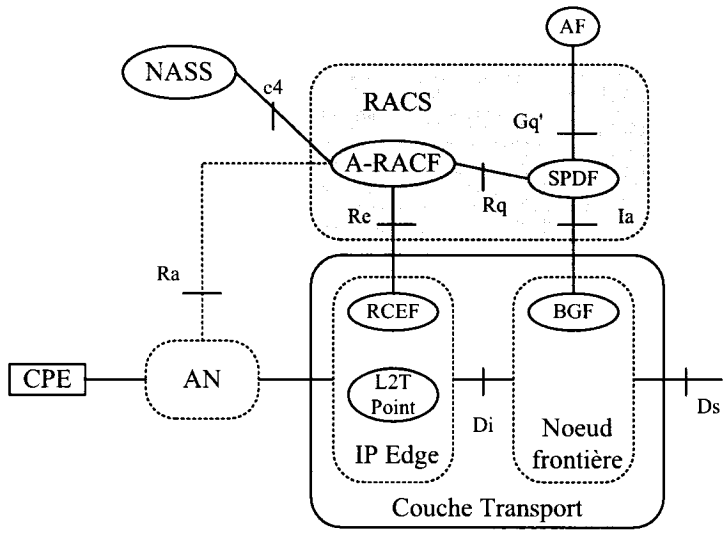

Figure 2.2 Architecture fonctionnelle d'un RACS

Le RACS contient:

- Le SPDF (*Service Policy Decision Function*) qui sert à évaluer les requêtes de connexions en fonction des politiques appliquées par le fournisseur de service. Si la décision est positive, la requête est acheminée vers le A-RACF avec la QdS demandée. Chaque fournisseur de service a un SPDF.
- Le A-RACF (*Access-Resource and Admission Control Function*) se charge d'abord de vérifier si la requête émanant du SPDF est conforme aux politiques du réseau d'accès. Ensuite, le A-RACF effectue les opérations nécessaires pour attribuer la QdS désirée et la réponse est retournée au SPDF.
- Le BGF (*Border Gateway Function*), situé à la frontière entre le réseau d'accès et le réseau cœur, permet l'application de politique sous le contrôle du SPDF. Il

effectue les fonctions de NAP(T). Il se charge par ailleurs d'effectuer certaines opérations de QdS comme le marquage des paquets sortants et entrants du réseau cœur ou la requête de bande passante dans le réseau cœur.

- Le RCEF (*Resource Control Enforcement Point*) se charge d'appliquer les décisions prises par le A-RACF. Il se charge d'autoriser le passage des paquets et de les marquer selon les critères du réseau d'accès. Il se situe au niveau du premier nœud IP (IP Edge) du réseau d'accès.

Dans cette architecture, l'interface R_a n'est pas définie et la QdS dans la portion de réseau avant le IP Edge n'est pas prise en compte. Cela pose un problème quand on parle de QdS de bout en bout. En effet, si une session requiert un certain niveau de QdS que cette portion de réseau ne peut fournir, la qualité de la communication s'en trouvera affectée.

2.1.3 MSF

Le MSF est un regroupement d'entreprise dont le but est de développer une architecture multiservices à partir de standards existants. Un de leurs objectifs fondamentaux est l'interopérabilité des technologies utilisées. La Figure 2.3 présente l'architecture de référence du MSF NGN.

Voici un résumé des fonctionnalités de certains éléments (MSF, 2005a):

- le TG (*Trunking Gateway*) et le SG (*Signalling Gateway)* permettent l'interconnexion avec les réseaux utilisant la signalisation *Signaling System 7* (SS7) et tous les réseaux à multiplexage temporel ;
- la passerelle d'accès permet de raccorder directement les téléphones classiques ne supportant pas SIP dans le réseau de l'opérateur ;
- les routeurs frontières acheminent le trafic IP vers le réseau du fournisseur d'accès. Ils effectuent aussi un contrôle d'accès pour la QdS ;
- le BM (*Bandwidth Manager)* est l'élément central de la gestion de la QdS. Il est responsable de l'allocation et de la libération des ressources. Il permet l'accès aux ressources pour les flots individuels à partir de politiques qui sont appliquées par

les routeurs frontières (MSF, 2005c). Il répond aux requêtes de ressources effectuées par les agents d'appel. Typiquement, il y a un BM par réseau. On retrouve donc des BM dans l'accès et dans le réseau cœur. Les réservations peuvent suivre le modèle *IntServ* dans les réseaux d'accès et *DiffServ* dans le réseau cœur ;

- les SBG (*Session Border Gateway)* permettent de contrôler les sessions en cours au niveau de la sécurité et de la translation d'adresses notamment. Il y a plusieurs types de SBG. Il y a ceux qui servent à contrôler les sessions entre deux fournisseurs et qui se retrouvent dans le réseau cœur (SBG-NC: SBG-*Network Core*). Un autre type de SBG se retrouve au niveau des routeurs frontières pour fournir certaines fonctions comme la sécurité (SBG-NE: SBG-*Network Edge*). Un troisième type, dont les fonctionnalités n'ont pas encore défini, pourrait être installé du côté de l'usager (SBG-CE: SBG-*Customer Edge*) ;
- l'agent d'appel initie et détruit les sessions. C'est l'élément qui demande les ressources au BM. Il se charge aussi des décisions de routage au cas où il faudrait contacter un autre agent d'appel. Il garde les informations relatives à la facturation. Chaque terminal usager est assigné à un agent d'appel qui se chargera d'invoquer les services pour l'usager.

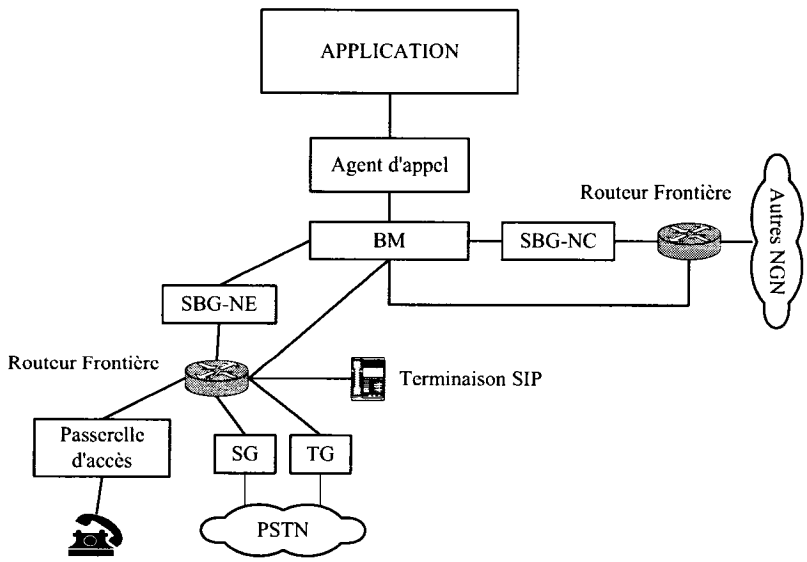

Figure 2.3 Architecture fonctionnel d'un réseau MSF NGN

Sur plusieurs plans, les architectures de TISPAN et de MSF sont similaires. Toutefois, MSF ne tient pas compte de la possibilité d'avoir des réseaux d'agrégation de niveau 2 dans le réseau d'accès avant de rencontrer le premier routeur IP. La QdS reliée à cet aspect n'est donc pas prise en compte alors que TISPAN mentionne clairement la présence d'un réseau d'agrégation de niveau 2 dans son architecture.

2.1.4 PacketCable

Comme nous l'avons mentionné au chapitre 1, l'objectif premier du réseau de PacketCable est de fournir des services multimédia à commutation de paquets IP sur le réseau existant de câbles en utilisant le protocole DOCSIS (PacketCable, 2005a). Une des motivations fondamentales de la mise en place de cette architecture réside bien entendu dans l'émergence du protocole IP pour le transport de différents types de services à QdS variable. Par ailleurs, l'existence des réseaux de câbles et les performances du protocole DOCSIS en matière de QdS réduisent les investissements nécessaires pour la mise en place de cette architecture. Toutefois, les spécifications de PacketCable ne tiennent compte que de IPv4. La Figure 2.4 présente l'architecture de référence de PacketCable.

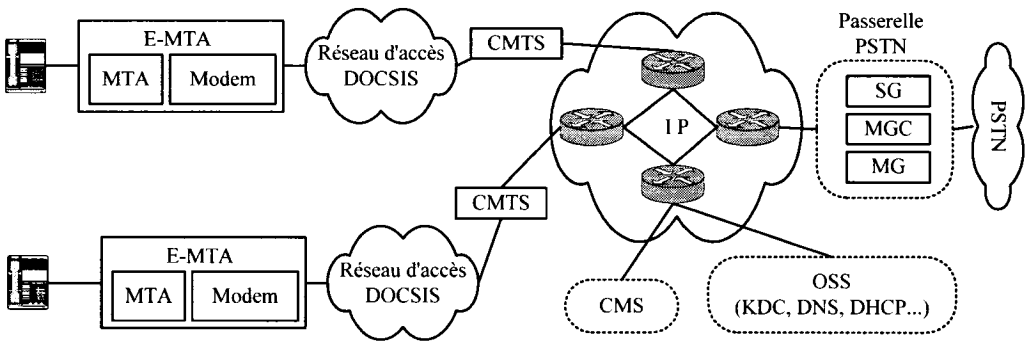

Figure 2.4 Architecture fonctionnelle d'un réseau PacketCable

L'architecture est composée de trois types de réseau:

- le réseau téléphonique public commuté (PSTN) ;
- le réseau IP qui permet d'interconnecter les différents réseaux d'accès et les éléments fonctionnels nécessaires pour la signalisation et les services ;

- les réseaux d'accès DOCSIS qui permettent d'offrir un accès rapide, sûr et sécuritaire aux usagers en plus des fonctions de QdS.

Les fonctionnalités des différents éléments de l'architecture sont expliquées dans la référence PacketCable (2005a). Au niveau de la QdS, la norme DOCSIS 1.1 définit un ensemble de fonctionnalités pour la gestion des ressources (PacketCable, 2005b). DOCSIS fournit un service de flot définit comme un service de transport de la couche MAC offrant un acheminement unidirectionnel des paquets et les fonctions de lissage, politiques et priorisation du trafic. DOCSIS permet de fournir de la QdS à des connexions statiques mais aussi a des connexions dynamiques. Les messages pour ajouter, changer ou retirer un flot sont émis par le modem ou le CMTS. En conclusion, le réseau de PacketCable est intéressant mais il se limite aux réseaux d'accès DOCSIS.

Maintenant que nous avons présenté les NGN, nous allons faire un tour d'horizon de la technologie Ethernet appliquée aux réseaux métropolitains.

2.2 MetroEthernet

MetroEthernet consiste en l'utilisation de la technologie Ethernet pour fournir des services de télécommunications dans les réseaux métropolitains. L'utilisation de MetroEthernet implique plusieurs avantages pour les opérateurs. Premièrement, Ethernet est une technologie mature et stable qui est présente dans plus de 90% des LANs. Il y a donc un avantage à conserver le format des données d'un LAN à l'autre, réduisant ainsi les coûts liés à la conversion des données et les erreurs pouvant en découler. Ensuite, les équipements Ethernet sont moins dispendieux que les équipements SONET par exemple. Par ailleurs, *Optical Ethernet* offre une grande largeur de bande ainsi qu'une portée des liens de connexions plus importante et, comparativement à SONET qui a été optimisé pour la voix, Ethernet se prête mieux à la nature irrégulière du trafic de données. Ethernet est aussi plus flexible quant à la granularité de la bande passante à offrir aux usagers et permet une meilleure utilisation des ressources par rapport au multiplexage temporel de

SONET par exemple. Dans ce paragraphe, nous parlerons d'Ethernet en général sans faire référence au médium de transmission.

2.2.1 Fonctionnement de MetroEthernet

Un réseau Ethernet est constitué de commutateurs reliés entre eux. Les commutateurs ont la particularité d'apprendre l'association entre l'adresse MAC d'une machine et le port par lequel celle-ci peut être rejointe, limitant ainsi les domaines de collisions. Pour pouvoir acheminer les données dans un grand réseau et éviter à chaque commutateur de diffuser les informations d'un groupe particulier d'utilisateurs sur tous ses ports, le concept de LAN virtuel (VLAN) a été proposé. Plusieurs machines qui sont sur des ports différents peuvent appartenir au même VLAN. Sur le plan topologique, un VLAN est un sous-ensemble d'un arbre de recouvrement. La norme IEEE 802.1Q (IEEE, 2003) introduit un entête ou étiquette VLAN de 4 octets qui se place entre l'adresse source et le champ Ethertype de l'entête Ethernet. Toutes les étiquettes VLAN sont communément appelées Q-TAG mais sont différentiées selon leur usage. Lorsqu'on utilise une seule étiquette comme le préconise le standard 802.1Q, on parle de C-TAG qui définit un C-VLAN. La Figure 2.5 présente la C-TAG, les champs en gris représentant les champs de la trame de base Ethernet.

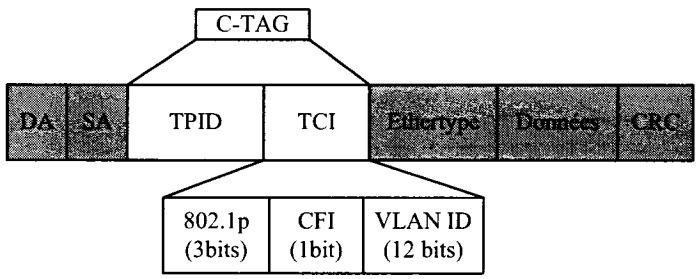

Figure 2.5 Trame Ethernet avec une étiquette VLAN

La C-TAG est composée d'un TPID de 16 bits servant à identifier le protocole, ainsi que d'un TCI dont les éléments les plus importants sont l'identifiant du VLAN du client de 12 bits (C-VID) et 3 bits de priorités (Chiruvolu et al, 2004).

Les VLANs permettent de séparer les domaines de diffusions. En effet, en cas de *broadcast* ou de *multicast*, la diffusion n'est effectuée que vers les machines appartenant au VLAN considéré. Cela permet d'améliorer l'utilisation de la bande passante. En plus de cela, les commutateurs apprennent les associations adresses MAC-port des machines afin de ne faire suivre les données que vers un port donné. Il est à noter qu'un réseau est limité à 4096 VLANs, ce qui peut poser problème dans le cas d'un réseau métropolitain.

Pour pallier le problème d'évolutivité lié au nombre de VLAN, IEEE a proposé les standards IEEE 802.1ad et IEEE 802.1ah (Elangovan, 2005, Botroff, 2005, IEEE, 2005). IEEE 802.1ad, connu aussi sous l'appellation Q-in-Q, consiste à utiliser deux étiquettes VLAN. Dans le cadre du IEEE 802.1ad, entre l'adresse MAC source et l'Ethertype, on retrouve une étiquette client, la C-TAG, et l'étiquette du fournisseur de service, la S-TAG. La Figure 2.6 illustre la trame Ethernet lorsque le standard IEEE 802.1ad est utilisé. Ces deux étiquettes sont aussi constituées d'un TPID et d'un TCI comme le C-TAG.

Figure 2.6 Trame Ethernet 802.1ad (QinQ)

Comme pour le C-TAG, le S-VID du S-TAG comporte 12 bits. Les S-TAG définissent un S-VLAN. Le fournisseur de service ne modifie pas le C-TAG et l'acheminement des trames dans le réseau du fournisseur se fait à partir du S-TAG qui est extrait des tables de concordances entre les C-VID et les S-VID contenus dans les commutateurs frontières. Ainsi, un client qui possède plusieurs V-LAN pourra conserver son C-TAG intact. Par ailleurs, un fournisseur de service peut choisir de faire une association un pour un entre les C-VID et les S-VID ou transporter plusieurs C-VID avec un même S-VID. Une approche intéressante aussi est d'associer un S-VID à chaque client, lui permettant ainsi de connecter ses différents sites. Il est aussi possible de transporter des trames provenant des clients sans C-TAG. Dans ce dernier cas, la trame n'aura que la S-TAG comme étiquette. En fait, le 802.1ad peut être vu comme une

généralisation du 802.1Q. Bien qu'utile, cette méthode pose un problème d'évolutivité car le fournisseur de services est limité à 4094 services Ethernet définis par les S-VLAN. Toutefois, le IEEE 802.1ad se prête plus à des réseaux métropolitains.

Par ailleurs, le *Metro Ethernet Forum* (MEF) a été mis en place pour normaliser le domaine du MetroEthernet. La phase I des travaux du MEF a consisté entre autres à définir le fonctionnement et les services qu'offrent le MetroEthernet (MEF, 2004A). En théorie, ces réseaux pourraient fonctionner en mode *broadcast* comme les réseaux Ethernet classiques mais cela pourrait entraîner une surcharge des liens dans le réseau. Le MEF définit donc une connexion virtuelle Ethernet (EVC). Une EVC peut se faire entre deux nœuds ou entre un ensemble de nœuds. Chaque nœud d'une EVC peut être considéré comme étant dans un réseau privé virtuel de niveau 2. Chaque nœud frontière garde une table de correspondance entre les VLAN ID du réseau et les EVC afin d'acheminer correctement les trames.

Lorsque l'on veut utiliser le concept de VLAN pour l'acheminement des trames dans un réseau de plus grande envergure (WAN), le problème lié au nombre de VLAN disponible ressurgit (Elangovan, 2005, Ali *et al.*, 2005a). Le groupe IEEE 802.1ah a proposé un standard et un format de trame qui est représenté à la Figure 2.7.

Figure 2.7 Trame Ethernet 802.1ah

Dans cette trame, on retrouve l'intégralité de la trame IEEE 802.1Q à l'exception du contrôle d'erreur qui est modifié. La trame IEEE 802.1ah inclue une étiquette de services étendus appelé I-TAG. L'opérateur du réseau dorsal effectue une correspondance entre le S-VID de la trame IEEE 802.1ad qui arrive et le I-SID de la trame IEEE 802.1ah qui va circuler sur le réseau. Un I-SID identifie donc un seul S-VLAN sur le réseau dorsal. Le S-VID du réseau d'origine et celui du réseau de destination peuvent cependant différer. Par ailleurs, les trames sont commutées à partir du B-TAG qui fonctionne identiquement au C-TAG mais au niveau de la dorsale. Les B-TAG définissent donc un B-VLAN. Enfin,

cette trame contient aussi les adresses destination et sources (B-DA et S-DA) qui identifient les nœuds du réseau dorsal. L'ajout de nouvelles adresses MAC répond au besoin de limiter les tables d'acheminement dans les commutateurs du réseau cœur. En effet, ces commutateurs ne doivent qu'apprendre les adresses MAC des commutateurs frontières.

La Figure 2.8 est un exemple de réseau hiérarchique qui pourrait utiliser Ethernet de bout en bout avec une combinaison de 802.1Q au niveau des réseaux clients, de 802.1ad dans l'accès du fournisseur et de 802.1ah sur la dorsale. Ce genre de réseau peut relier différents sites appartenant au même client (A, B, C et D sur la Figure 2.8).

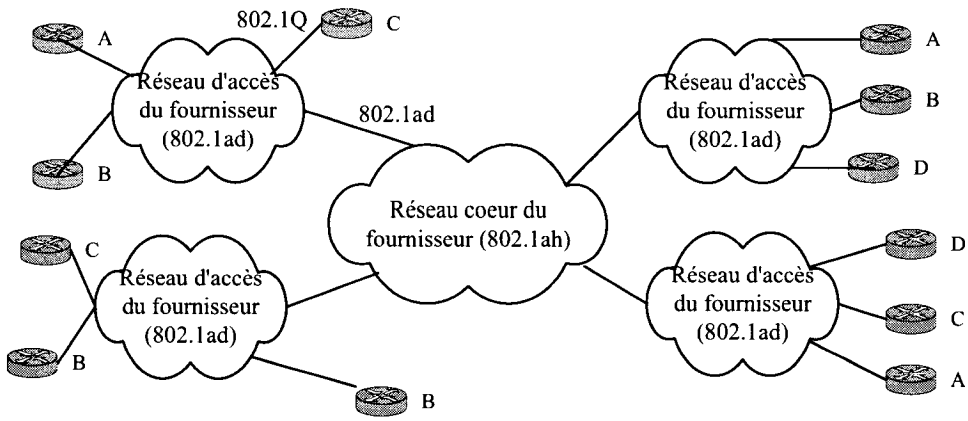

Figure 2.8 Réseau hiérarchique utilisant Ethernet de bout en bout

Après avoir présenté les réseaux MetroEthernet, nous allons maintenant traiter des aspects de QdS reliés à cette technologie.

2.2.2 Ethernet et la qualité de service

Ethernet de base ne définit pas de procédé de QdS. Il fonctionne en *broadcast* avec la même priorité pour chaque trame. La norme IEEE 802.1p (IEEE, 2004) introduit un concept de priorité qui est limité à huit différents niveaux. Les priorités peuvent être codées sur 3 bits inclus dans l'étiquette IEEE 802.1Q. Les seuils des paramètres qui déterminent les niveaux de priorité peuvent varier d'un réseau à un autre. Ces niveaux

priorités sont associées à différents types de trafic. 802.1p définit 7 types de trafics qui sont:

1. trafic d'arrière plan (*Background*): activités permises sur le réseau mais qui ne doivent avoir aucune incidence sur l'utilisation du réseau par les autres usagers et les applications ;
2. meilleur effort: service offert selon la bande passante disponible non réservée ;
3. excellent effort: service meilleur effort offert aux plus importants clients ;
4. charge contrôlée: importantes applications d'affaire sujettes à un contrôle d'admission et une forme de débit garanti ;
5. vidéo: caractérisée par un délai inférieur à 100 ms ;
6. voix: caractérisée par un délai maximum de 10 ms et une gigue maximale ;
7. commande de réseau: trafic nécessaire au maintien de l'infrastructure réseau et qui doit impérativement arriver à destination.

Le tableau 2.1 présente une concordance entre le niveau de priorité usager et les types de trafics définis par 802.1p.

Tableau 2.1 Priorité usager et type de trafic

Priorité Ethernet	Exemple d'application
1	Arrière plan
2	Non défini
0 (Défaut)	Meilleur effort
3	Excellent effort
4	Charge contrôlée
5	Vidéo
6	Voix
7	Commande de réseau

Par ailleurs, le nombre de files d'attente disponibles à un port, qui correspond au nombre de classes possibles, ne correspond pas toujours au nombre de priorités qui existent. Le tableau 2.2 présente la correspondance définit par 802.1p pour ce cas.

La priorisation est un aspect très important pour la QdS qui est un élément majeur pour certaines applications. Si le nombre des classes de trafics pour un port est peu élevé, l'accent est mis sur le délai offert aux trafics. Plus le nombre de classes augmente, plus la différenciation est meilleure et il est aussi possible de mettre l'accent sur le débit accordé aux différents trafics.

Tableau 2.2 Correspondance Priorité/Classe en fonction du nombre de classes

Priorité d'usager	Nombre de classes disponibles pour un port							
	1	2	3	4	5	6	7	8
0 (Défaut)	0	0	0	1	1	1	1	2
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1
3	0	0	0	1	1	2	2	3
4	0	1	1	2	2	3	3	4
5	0	1	1	2	3	4	4	5
6	0	1	2	3	4	5	5	6
7	0	1	2	3	4	5	6	7

Les huit classes permettent de fournir une qualité de service différenciée comme *DiffServ*. Toutefois, si Ethernet doit être utilisé dans les réseaux métropolitains et fournir des services tels que la voix et la vidéo, le problème de la concordance au niveau de la QdS apparaît. Il ne faudrait pas que le passage dans un domaine Ethernet apporte une dégradation notable des performances de bout en bout. Pour cela, il faut établir une relation entre les 3 bits de l'entête VLAN et les 6 bits du champ DSCP de *DiffServ*, les 8 bits du champ ToS (*Type of Service)* de IP ou les mécanismes de signalisations RSVP. La concordance avec RSVP semble encore plus difficile parce que celui-ci se base sur *IntServ* et fait une signalisation par flot alors que IEEE 802.1Q se base sur une approche d'aggrégation similaire à *DiffServ*. De plus, Ethernet ne permet pas d'identifier des flots individuels à partir de son entête.

Une des grandes lacunes d'Ethernet réside dans la rareté des architectures de QdS appropriées à ce type de réseau. Comme Ethernet utilise une QdS de type *DiffServ* avec les 3 bits de priorité, les mêmes problèmes de congestion liés à l'absence de contrôle d'admission pour les demandes de connexions dynamiques dans les réseaux IP *DiffServ* sont rencontrés. Toutefois, certaines solutions ont été apportées pour permettre d'utiliser RSVP sur les réseaux locaux IEEE 802. Nous allons en décrire quelques-unes.

Subnet Bandwidth Manager (SBM)

Yavatkar *et al.* (2000) et Vogt (2002) proposent des solutions permettant d'utiliser une approche *IntServ* sur un réseau Ethernet. Les auteurs introduisent un gestionnaire de sous-réseau, le SBM, qui sert au contrôle d'admission dans chaque segment de niveau 2 désigné par domaine L2.

Lorsqu'une requête de réservation de type *IntServ* est envoyée sur le domaine L2, elle est acheminée vers le SBM. Ce dernier, qui connaît les capacités du réseau à l'instant de la requête, évalue la possibilité d'accepter ou non la réservation par rapport aux différents niveaux de priorité. Un flot est accepté pour un niveau de priorité donné si son addition ne dégrade pas les performances de tous les flots ayant la même priorité au delà des seuils de performances assignés à cette priorité. Si le flot est accepté, le SBM achemine la réservation vers les commutateurs et la machine destination si celle ci se trouve dans son réseau. Le niveau de priorité est retourné à l'émetteur par le SBM, qui devra l'insérer dans ses trames. Cette approche est bien définie pour RSVP mais fonctionnerait avec n'importe quel protocole de réservation basée sur *IntServ*.

Le SBM peut être implémentée de façon centralisée ou distribuée dans chaque commutateur. Toutefois, les fonctions de lissage de trafic et d'application des politiques sont extérieures à cette architecture. En plus, la solution SBM est définie pour les réseaux locaux et les auteurs ne traitent pas de son applicabilité aux réseaux MetroEthernet. Aussi, la solution implique que toutes les données transitent par le SBM, ce qui en fait un goulot d'étranglement qui peut être problématique en cas de pannes.

MetroEthernet Forum (MEF)

Le MEF utilise les 3 bits de l'étiquette VLAN pour les classes de service. Un EVC peut transporter des trames associées à une ou plusieurs classes de service (CdS). Les CdS sont caractérisées par les paramètres de délai, de gigue et de taux de perte de paquets. Des profiles de QdS sont attribuées soient aux EVC, soient aux classes dans les EVC. Pour réguler le trafic à l'entrée du réseau, une variante de l'algorithme du seau à jeton percé est utilisée (MEF, 2004a).

Le MEF définit une architecture de gestion de réseau basée sur des *Element Management System* (EMS) et un *Network Management System* (NMS). Dans un réseau, chaque sous-réseau ou domaine de flot est géré par un EMS. Les EMS sont reliés à un NMS (MEF, 2004b). La Figure 2.9 présente un exemple de l'architecture préconisée par MEF.

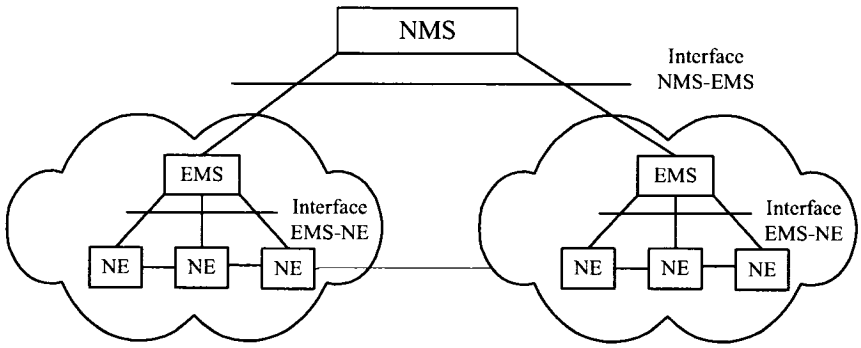

Figure 2.9 Architecture de gestion préconisée par le MEF

Ce type d'architecture permet au NMS d'avoir une vue globale du réseau et il permet aussi d'administrer différentes technologies. Un des avantages de cette vue globale est de pouvoir gérer les connexions de bout en bout. Dans le fonctionnement, les requêtes de connexions sont acheminées au NMS. Celui ci détermine les domaines de flots qui seront impliqués et adresse une requête de connexion à chaque EMS avec les informations relatives à la QdS. Lorsque le EMS reçoit la requête, il initie le processus de configuration de ressources en vigueur dans son domaine pour satisfaire à la demande. Il répond ensuite au NMS qui décide de l'acceptation ou non de la requête. Il faut cependant noter que le MEF ne décrit pas comment le EMS effectue la réservation des ressources.

Toutefois, le MEF a produit un ensemble de requis pour la définition d'une architecture dynamique de gestion de QdS (MEF, 2005) mais aucune proposition concrète n'a été effectuée.

Solutions de l'ITU-T

L'ITU-T a produit deux propositions définissant une architecture de gestion pour un réseau Ethernet à grande échelle. La première proposition concerne un réseau d'accès similaire à celui de TISPAN (voir Figure 2.1) intégrant Ethernet comme réseau de transport L2 (ITU-T, 2005b). La solution proposée considère la portion IP et la portion Ethernet comme un tout. Le RACF est informé de la topologie physique et logique du réseau grâce aux interfaces qu'ils partagent avec chaque nœud du réseau. Il est chargé du contrôle d'admission et de s'assurer qu'aux extrémités du réseau d'accès, les flots respectent les seuils fixés. Le RACF se charge du routage, des réservations et de l'assignation des classes de services aux flots. On peut reprocher à cette proposition le fait de ne pas décrire ce qui se passe à l'intersection entre la portion Ethernet et la portion IP. Les auteurs ne mentionnent pas le fait que le contrôle aux extrémités soit fait par flots individuels ou par agrégats. Dans le cas de flots individuels, leur identification au niveau Ethernet poserait un problème.

La deuxième proposition suppose un réseau purement Ethernet comportant des réseaux d'accès et un réseau cœur (ITU-T, 2006). Les auteurs utilisent MPLS dans le réseau cœur. Une étiquette est ajoutée à la trame Ethernet pour transporter des informations sur la priorité, la bande passante et la sécurité notamment. Les commutateurs frontières du fournisseur de service permettent de classifier les trames et de vérifier la conformité du trafic aux ententes établies. Par ailleurs, si le trafic associé à une classe donnée excède les exigences, les trames peuvent être jetées ou marquées candidate au rejet (DE: *discard eligible*). Le taux de trames DE pour une classe est limitée pour ne pas nuire aux classes moins prioritaires.

Cette proposition étend *DiffServ* à Ethernet en définissant trois types de comportements par nœud (PHB):

- E-EF (*Ethernet Expedited Forwarding*): cette classe correspond aux services avec des contraintes sévères de délai et de perte de paquets ;
- E-AF (*Ethernet Assured Forwarding*): définit un ensemble de classes avec un niveau de précédence pour le rejet des trames. Ces classes garantissent un minimum de largeur de bande mais le délai n'est pas borné ;
- DF (*Ethernet Default Forwarding*): cette classe correspond aux services de meilleur effort.

Le tableau 2.3 donne un exemple de l'utilisation des bits de priorité pour définir les différents PHB. Les auteurs introduisent un élément qui sert à faire le contrôle d'admission mais les processus de réservation et de relâchement de ressources ainsi que les interfaces du réseau ne sont pas clairement définis.

Tableau 2.3 PHB Ethernet

Ethernet (p-bits)	E-PHB	Exemple d'application
111	E-EF	Service d'urgence
110	E-AF31	Voix
101	E-AF32	Vidéoconférence
100	E-AF21	Commande vocale
011	E-AF22	Données critiques
010	E-AF11	Vidéo Streaming
001	E-AF12	Données (Importance moyenne)
000	E-DF	Données (Meilleur effort)

Comme nous l'avons déjà mentionné, Ethernet utilise une approche *DiffServ* pour la QdS. Or, à l'origine, *DiffServ* ne comprenait pas de mécanismes de contrôle d'admission et de réservation de ressources dynamiques. Certaines solutions ont été apportées pour remédier à cela, parmi lesquelles *Resource Management in DiffServ* (RMD) que nous allons décrire. Précisons qu'à l'heure actuelle, cette solution ne s'applique qu'aux réseaux de type IP.

RMD

RMD est une technique pour ajouter un contrôle d'admission et des fonctions de préemption aux réseaux *DiffServ* (Westberg *et al.*, 2002). RMD permet à des éléments extérieurs au domaine *DiffServ* de réserver dynamiquement des ressources auprès des routeurs frontières. Le nœud entrant du domaine *DiffServ* classifie le trafic et signale la réservation de ressources sur tout le chemin tandis que le nœud sortant achemine la requête originale vers la destination finale. Il existe deux types de protocoles RMD: les protocoles de réservation par domaine (PDR) et les protocoles de réservation par saut (PHR). Les PHR sont internes au domaine *DiffServ* tandis que les PDR servent à la réservation entre les routeurs frontières des domaines impliqués. Le même protocole (RSVP ou NSIS par exemple) peut être utilisé comme PHR et PDR.

Avec RMD, les nœuds peuvent conserver des états pour chaque flot, conserver des états pour chaque classe de trafic (états réduits) ou encore ne pas conserver d'états. La conservation d'états pour chaque flot peut se heurter au problème d'évolutivité car les états, incluant les états réduits, doivent être périodiquement rafraîchis. L'absence d'états ou la conservation d'états réduits règlent le problème d'évolutivité mais le niveau de service offert aux flots individuels peut s'en trouver dégrader. Pour pallier cela, RMD propose de conserver les états pour chaque flot au niveau des routeurs frontières et de ne conserver que des états réduits ou pas d'états du tout dans les routeurs internes au domaine *DiffServ*. Ainsi, les routeurs frontières exercent un contrôle d'admission et s'assurent que les paramètres des flots restent dans les limites fixées tandis que les routeurs internes peuvent acheminer plus rapidement les paquets à cause du peu de traitements à effectuer.

RMD définit deux modes de contrôle d'admission: le contrôle d'admission de connexion (CAC) basé sur les mesures et celui basé sur les réservations (Bader *et al.*, 2006). Le CAC basé sur les mesures (CACM) se fait en ne conservant aucun état relatif aux flots dans les routeurs internes. Chaque routeur interne maintient deux états pour chaque classe: le premier état mesure le volume courant de trafic et le deuxième état conserve le seuil maximal à ne pas dépasser. À l'arrivée d'une requête, les routeurs

intermédiaires acceptent la requête en fonction de l'état actuel du trafic. Une variante consisterait à envoyer des paquets tests au débit désiré pour voir comment le réseau répondrait. Le CACM est apparenté aux méthodes sans états car on ne conserve aucun état relatif aux différents flots individuels.

D'autre part, le CAC basé sur les réservations (CACR) implique que chaque routeur interne conserve des états réduits c'est à dire des états relatifs à chaque classe qui sont mis à jour par des messages de rafraîchissements. Lors de l'arrivée d'une requête R, le routeur d'entrée du domaine *DiffServ* émet un message le long du chemin. Chaque routeur intermédiaire vérifie s'il peut ajouter, sans dépasser une certaine limite, la quantité de ressources demandée par R au total de la réservation de la classe de service à laquelle appartient R. Si oui, la requête se propage jusqu'à la destination. Pour annuler la réservation associée à R, les routeurs du chemin retranchent la quantité de ressources utilisées par R du total de la classe associée. CACR est une méthode plus sûre que CACM car elle fonctionne en fonction des réservations en cours et un flot qui atteint son pic pour une durée donnée ne gênera pas les autres réservations. Toutefois, CACM permet une meilleure utilisation des ressources car l'acceptation dépend de l'état courant du trafic.

La Figure 2.10 montre la réservation de ressources avec le protocole NSIS.

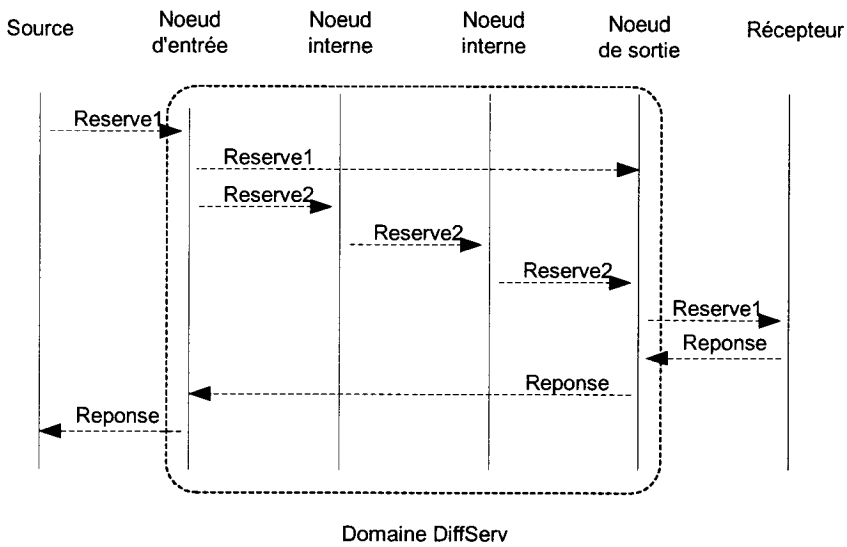

Figure 2.10 Réservation de ressource avec RMD

Une fois que le routeur entrant du domaine *DiffServ* reçoit le message *Reserve1* initial, il l'achemine au routeur de sortie. Le message *Reserve1* n'est pas traité par les routeurs internes. Le routeur d'entrée émet un message *Reserve2* qui lui sera traité par les routeurs internes pour l'installation de la réservation selon le mode de contrôle d'admission utilisé. Une fois que le routeur de sortie reçoit les deux messages, il vérifie qu'il y a concordance et que la réservation a été acceptée puis il propage le *Reserve1* vers la destination finale. Un message *Reponse* est émis pour confirmer la réservation. Enfin, RMD définit aussi des mécanismes pour gérer la congestion en marquant les paquets des flots qui subissent la congestion afin que les routeurs frontières puissent prendre une décision par rapport aux flots concernés. Comme nous le voyons, RMD est une technique intéressante dont certains aspects pourraient être appliqués aux réseaux Ethernet.

Autres solutions de QdS pour Ethernet

Chamas et al. (2005) proposent une architecture pour le contrôle d'admission des services Ethernet. L'architecture est basée sur un serveur pour le contrôle d'admission. Trois classes de services sont définies et un pourcentage maximal de la capacité de chaque lien peut être attribué à chaque classe. Le serveur connaît la topologie logique basée sur les arbres de recouvrement et le taux d'utilisation des liens par classe. Lors de l'arrivée d'une requête, le serveur recherche le plus court chemin en fonction de cette topologie logique. Les résultats obtenus permettent une meilleure utilisation des liens ainsi que l'acceptation d'un plus grand nombre de requêtes. Toutefois, les auteurs ne mentionnent pas si les requis de QdS de bout en bout des flots individuels sont satisfaits.

Par ailleurs, les standards 802.1ad et 802.1ah apportent quelques solutions au problème de qualité de service dans Ethernet. D'une part, avec le 802.1ad, l'utilisation d'une étiquette supplémentaire dans laquelle on retrouve un bit indiquant la possibilité de jeter la trame (*Drop Eligible bit*) permet de doubler le nombre de classes en introduisant un ordre relatif dans chaque classe. D'autre part, le 802.1ah utilise un label de taille minimale 20 bits. Ces solutions sont intéressantes mais elles rajoutent de l'information de contrôle dans les trames Ethernet réduisant ainsi le pourcentage de charge utile.

Multi-Protocol Label Switching (MPLS) et *Generalized* MPLS apparaissent aussi comme des avenues intéressantes pour l'acheminement de trames Ethernet à cause de leurs fonctionnalités d'ingénieries de trafic. Les sections suivantes décrivent MPLS et GMPLS ainsi que leur applicabilité à Ethernet.

2.2.3 MPLS et GMPLS

MPLS est un mécanisme de transport de données basé sur la commutation d'étiquettes de niveau 2. Contrairement au routage IP qui se base sur une analyse de l'en-tête IP, MPLS achemine l'information en fonction d'une étiquette. Cela permet d'accélérer le transfert des informations. Les protocoles de signalisation comme RSVP et *Constrained Routing-Label Distribution Protocol* (CR-LDP) permettent de réserver les ressources et de distribuer les étiquettes. GMPLS est une extension de MPLS-TE (*MPLS-Traffic Engeenering*) permettant de contrôler de manière coordonnée et efficace les différentes couches de réseaux (Projet AGAVE, 2004). GMPLS est un protocole de contrôle réparti permettant à toutes les couches de connaître la topologie du réseau.

GMPLS permet différents types de commutations qui sont hiérarchisées (paquets, couche 2, multiplexage temporel, longueur d'onde et fibre). Certains équipements évolués sont capables de commuter à plusieurs niveaux. Dans le cadre de GMPLS, la notion de *Label Switched Path* (LSP) est généralisée et s'applique à chacune des couches impliquées dans le réseau. Toutefois, un LSP doit débuter et terminer à des interfaces de même type.

Le label de MPLS et de GMPLS est un aspect important de ces technologies car il permet d'effectuer la commutation. La taille du label MPLS est de 4 octets tandis que le label généralisé de GMPLS peut être de taille variable selon le niveau de commutation. Par exemple, la commutation de cellules ATM peut se faire à partir des champs VPI et VCI de la cellule.

MPLS est une technologie prometteuse à cause des coûts et de l'ingénierie de trafic. Elle permet d'offrir des garanties de QdS à différents types de protocoles comme IP. De ce fait, la nécessité d'utiliser des technologies coûteuses comme ATM pour garantir la

QdS s'amenuise. On s'oriente de plus en plus vers des réseaux métropolitains ou dorsaux totalement MPLS ou GMPLS. Cela permet aussi d'avoir un plan de contrôle unifié pour transporter différents types de services.

2.2.4 Ethernet sur les réseaux MPLS et GMPLS

Comme nous l'avons vu, MPLS, GMPLS et Ethernet s'alignent pour être des technologies dominantes pour la commutation dans les réseaux du futur. Pour différencier MPLS et GMPLS lors de l'utilisation d'Ethernet, il faut mentionner que l'utilisation de MPLS implique un niveau d'encapsulation supplémentaire car les opérateurs doivent d'abord gérer le réseau au niveau des LSP de paquets avant de gérer les connexions Ethernet tandis que le GMPLS permet de gérer directement les connexions Ethernet (L2LSP) tout en leur assurant les mêmes garanties que MPLS. GMPLS permet de mettre en place automatiquement les L2LSP et donne un plan de contrôle unifié aux opérateurs leur permettant d'avoir une information globale sur l'état du réseau (Papadimitriou *et al.*, 2005).

Certains équipementiers offrent déjà des commutateurs avec des fonctionnalités GMPLS. Toutefois, même si GMPLS permet la commutation de niveau 2, le cas spécifique de la commutation Ethernet avec GMPLS n'a pas encore fait l'objet d'un standard internationalement reconnu. En effet, pour transporter des trames Ethernet sur un réseau MPLS ou GMPLS, il convient de définir un format de label ainsi que les opérations qui seront effectués par les commutateurs.

Une chose importante est que toute solution proposée puisse fonctionner avec les réseaux Ethernet classiques. On désigne l'utilisation de Ethernet avec GMPLS par ELS pour *Ethernet Label Switching* (Anderson et Papadimitriou, 2005). Un des objectifs du ELS est d'utiliser un label inséré dans l'entête Ethernet. Plusieurs solutions ont été proposées pour l'utilisation d'Ethernet sur des réseaux MPLS ou GMPLS et quatre grands groupes de solutions se dégagent. Nous allons présenter sommairement ces solutions (Anderson et Papadimitriou, 2005).

2.2.4.1 Utilisation classique de MPLS

Une solution possible et fonctionnelle préconise l'utilisation de l'étiquette MPLS qui est insérée entre l'entête Ethernet et l'encapsulation de niveau 3. Cette solution impliquerait cependant que les commutateurs ajoutent, retirent ou changent les étiquettes à chaque nœud car les étiquettes auraient une signification locale sur un lien donné. Par ailleurs, il faudrait aussi réécrire les adresses MAC sources et destination à chaque saut. Dépendamment de l'implémentation, il faudrait ajouter une étiquette MPLS en plus d'un Q-TAG. Cette solution pourrait être qualifiée de MPLS sur Ethernet.

Pour faire circuler des trames Ethernet sur un réseau MPLS, on parle aussi de Ethernet sur MPLS (EoMPLS). EoMPLS définit une connexion Ethernet point à point. Dans Aggarwal (2004) et Bryant et Pate (2005), les auteurs proposent le concept de *pseudo-wire* (PW). Un PW est un mécanisme permettant d'émuler les attributs essentiels d'un service de télécommunications comme *Frame Relay* ou ATM sur un réseau à commutation de paquets. Un PW peut interconnecter deux réseaux de technologies différentes. Il est nécessaire d'encapsuler les unités de données arrivant sur le réseau pour pouvoir les acheminer au travers d'un tunnel. Certaines autres opérations comme la signalisation spécifique, le séquençage ou le contrôle du délai peuvent être nécessaires.

MPLS est une technologie qui peut être utilisée pour la formation du tunnel. Particulièrement, un PW MPLS peut transporter des trames de niveau 2 (Martini *et al.*, 2006). Cette technologie nécessite l'utilisation de deux étiquettes distinctes. La première étiquette, appelée étiquette de tunnel ou de chemin, sert à commuter la trame à travers le réseau MPLS. La deuxième étiquette, l'étiquette de circuit virtuel, sert en fait à identifier la technologie qui est émulé sur le PW. Martini *et al.* (2005) définissent le fonctionnement spécifique de Ethernet avec les PW dont la Figure 2.11 est un exemple.

Figure 2.11 Encapsulation Martini (Ethernet dans Pseudo-Wire)

Les PW peuvent transporter des étiquettes VLAN mais celles qui ont une valeur locale sont retirées. L'ITU a aussi fait une proposition pour le transport de trames

Ethernet sur un réseau MPLS. Cette proposition est similaire au PW (ITU-T, 2005a). La trame Ethernet est encapsulé par trois éléments de 4 octets chacun. Le premier élément est appelé étiquette de transport et sert à la commutation MPLS. Le deuxième élément, l'étiquette d'interfonctionnement, permet d'associer un LSP à une connexion Ethernet. Enfin, les indicateurs communs, qui sont facultatifs, peuvent servir pour la fragmentation et le séquençage des trames.

D'autre part, le besoin de réseaux privés à grande échelle se fait de plus en plus ressentir par exemple pour qu'une même entreprise puisse transporter de façon sécuritaire ses données d'un site à un autre. Les VPN (*Virtual Private Network)* sont un moyen pour un client de construire un tel type de réseau virtuel. On retrouve des VPN de niveau 3 (L3VPN) et des VPN de niveau 2 (L2VPN). Les L3VPN sont basé sur IP tandis que les L2VPN permettent de transporter d'autres types de technologie de niveau 3. L'évolution de EoMPLS a débouché sur VPLS (*Virtual Private LAN Service).* Un VPLS est l'émulation d'un LAN Ethernet sur un réseau MPLS en utilisant des PW (Ali *et. al*, 2005b). En fait, chaque routeur frontière MPLS faisant partie du VPLS établit des PW bidirectionnels vers chacun des autres routeurs frontières MPLS appartenant au VPLS. Ainsi, le VPLS se comporte comme un commutateur dont les ports sont les routeurs frontières MPLS du VPLS. Le fonctionnement est identique à celui d'un LAN Ethernet avec l'acheminement basé sur l'apprentissage des adresses. Les solutions basées sur EoMPLS sont intéressantes mais elles impliquent l'ajout d'une étiquette supplémentaire en plus de nécessiter des traitements particuliers pour le formatage des données avant l'encapsulation dans le PW.

2.2.4.2 Utilisation du Q-TAG

Dans cette approche, la commutation GMPLS se fait à partir du Q-TAG. Elle présente l'avantage que la commutation à partir du numéro de VLAN est supportée par la plupart des commutateurs. Toutefois, l'espace des étiquettes dans ce cas est restreint à 12 bits et l'évolutivité n'est pas garantie. En plus, si la définition du VID est modifiée de la sorte, il devient impossible d'en faire de façon simultanée l'utilisation classique prévue

par 802.1Q. En effet, sur un point de vue conceptuel, le VID tel que définit par 802.1Q ne peut servir à faire de la commutation car il définit un réseau virtuel. Le commutateur ne peut prendre de décision par rapport au port spécifique de sortie à partir du VID qui ne sert qu'à limiter le domaine de diffusion. Par ailleurs, l'utilisation du VID comme étiquette de commutation impliquerait la désactivation de l'acheminement basée sur l'adresse MAC destination.

2.2.4.3 Utilisation de l'espace de la MAC adresse

Dans le cadre de ELS, une solution évidente serait d'utiliser l'adresse MAC de destination comme étiquette de commutation surtout qu'elle est censée être unique. Si on conserve l'adresse MAC dans son format actuel, trois inconvénients majeurs apparaissent. Premièrement, un commutateur recevant, à partir de deux ports différents, deux trames avec la même adresse de destination ne pourrait les différencier par rapport à la QdS à fournir. Le deuxième problème relève du fait que si plusieurs connexions avec une QdS différentes existent entre deux stations, il est impossible de les différencier à partir des adresses MAC. Enfin, cette solution impliquerait la création d'un LSP pour chaque connexion existante sur le réseau. On perdrait ainsi les avantages liés à l'agrégation de trafics avec des requis de QdS similaires.

D'autres recherches ont été effectuées dans le cadre de l'utilisation de l'espace de l'adresse MAC. En particulier, elles se basent sur le fait que l'adresse MAC destination est un élément statique lors de l'acheminement d'une trame. Une adresse MAC est constituée de 3 octets qui servent à identifier le manufacturier (numéro OUI : *Organization Unique Identifier*) et de 3 autres qui servent à identifier une carte d'accès au médium d'un manufacturier donné. L'idée derrière cette approche serait de réserver un OUI pour l'ELS. Les 3 octets restant serviraient à définir l'étiquette. Jaihyung (2005) et Jaihyung (2006) proposent que les 24 bits soient entièrement utilisés afin de définir un espace d'environ 16 millions d'étiquettes. L'avantage d'avoir un si grand nombre d'étiquettes est qu'elles peuvent être conservées de bout en bout dans un réseau et ne sont pas juste significatives pour un lien. Il y a donc moins d'opérations à effectuer. Feuseu et

Cousin (2005) proposent de partitionner l'espace des 24 bits en deux ensembles de tailles variable: le premier ensemble représente une étiquette de chemin et le second, une étiquette d'hôte. L'étiquette de chemin a une signification locale par rapport aux liens et est changée à chaque commutateur tandis que l'étiquette d'hôte permet au commutateur de sortie de replacer la véritable adresse MAC de destination. L'étiquette d'hôte détermine aussi la limite du nombre de nœuds pouvant être dans le réseau. Dans le même genre de solutions, Zeng *et al.* (2005) introduisent une solution où l'adresse MAC destination est remplacée par des informations de commutations prenant en compte la longueur d'onde dans un réseau optique ainsi que le port de sortie. Toutefois, cette solution ne tient pas compte de la présence d'un numéro OUI et l'interopérabilité avec les systèmes actuels risque d'être difficile. Pour résumer, on peut dire que même si ce genre d'approches permet d'obtenir un espace important pour les étiquettes, elles rentrent en conflit avec la sémantique actuelle des réseaux Ethernet. Un commutateur classique ne saurait comment gérer une trame semblable. En plus, il faut réécrire l'adresse MAC destination avant de la transmettre au destinataire, ce qui peut être une source d'erreurs.

Fedyk et Allan (2005) proposent d'utiliser la combinaison de l'adresse MAC et du VID comme label. En effet, cette combinaison est unique. L'idée consiste à réserver un certain nombre de VID pour cette utilisation. Un L2LSP serait donc uniquement identifié par les 60 bits de la combinaison adresse MAC destination – VID réservé. C'est une solution intéressante hormis le fait qu'elle réduit l'espace des VLANs pouvant être utilisés comme le spécifie le standard 802.1Q. En plus de cela, la sémantique du VID est modifiée.

2.2.4.4 Utilisation d'un nouveau TAG

Comme nous l'avons mentionné à la section 2.2.2, le Q-TAG contient un TPID pour identifier le protocole. Cette approche vise à définir un nouveau TAG de 4 octets muni d'un TPID différent. Cela permettrait d'utiliser les 13 bits restants comme étiquette. Ainsi, il n'y aurait aucune modification de la sémantique des champs fonctionnels actuellement utilisés pour Ethernet. L'utilisation de l'adresse MAC demeure aussi inchangée.

Toutefois, cette méthode peut soit résulter en l'augmentation de la taille de l'entête si on ajoute le nouveau TAG à ceux existants, soit entraîner l'impossibilité d'utiliser les VLANs si le nouveau TAG se substitue à l'un de ceux qui sont déjà fonctionnels.

Après avoir présenté les liens entre MPLS, GMPLS et Ethernet pour l'acheminement des trames, nous allons traiter le CAC qui est important pour assurer une QdS acceptable dans tout réseau.

2.3 Routage avec qualité de service et contrôle d'admission

Dans la section 2.2.2 qui traite d'Ethernet et de la QdS, nous avons présenté des architectures de CAC mais nous n'avons pas traité le processus de décision en tant que tel. Nous détaillons cet aspect dans cette section.

Premièrement, il faut dire que le routage avec QdS consiste à trouver un chemin réalisable de la source à la destination en respectant certaines contraintes de QdS. Le contrôle d'admission des connexions (CAC), quant à lui, consiste à autoriser ou refuser des connexions en se basant sur les politiques en vigueur, l'état des ressources du réseau et les contraintes de QdS. Si le CAC n'est pas correctement effectué, le réseau pourrait autoriser l'accès à un trop grand nombre de connexions, surchargeant ainsi les liens et dégradant le niveau de QdS de certaines applications. Très souvent, le CAC se base sur le routage avec QdS pour prendre la décision d'acceptation mais le CAC peut aussi se baser sur les politiques et rejeter une connexion même s'il existe un chemin réalisable.

Les contraintes de QdS sont de deux genres : les contraintes locales (lien ou nœud) et les contraintes de chemins (bout en bout). La contrainte locale la plus utilisée est la contrainte de capacité. Le délai, la perte de paquets et la gigue sont des exemples de contraintes de chemins qui peuvent aussi être considérées localement.

Au niveau du routage, les algorithmes de Dijkstra et Bellman-Ford sont reconnus pour trouver des plus courts chemins dans un réseau (Ahuja *et al.*, 1993). Le routage avec QdS peut être statique ou dynamique. La création d'une topologie logique avec MPLS peut être vue comme un cas de routage statique (Chen *et al.* 1995, Burns *et al.* 2003,

Dias *et al.* 2003). Nous nous intéressons au routage dynamique qui semble plus approprié par rapport aux réseaux de prochaines générations. Kodialam et Lakshman (2000) proposent l'Algorithme de Routage avec Interférence Minimum (MIRA). Leur objectif est de router le trafic à travers un chemin qui interfère le moins possible avec les requêtes futures. Cela est effectué en conservant une liste des liens dont l'utilisation peut réduire le maximum du flot potentiel entre les autres paires de nœuds frontières. Toutefois, il faut mettre un bémol à cette approche car elle nécessite, pour chaque requête, le calcul du flot maximum pouvant circuler pour chaque paire origine destination, ce qui requiert un temps important. Par ailleurs, Capone *et al.* (2003) ont proposé une méthode de déviation de flot qui permet d'acheminer le trafic d'une même application sur différents chemins, ce qui peut introduire des problèmes de gigue. D'autre part, Widyono (1994) a modifié l'algorithme de Bellman-Ford pour tenir compte des contraintes mais cet algorithme entraîne une recherche fastidieuse du chemin optimal.

Le routage avec plusieurs contraintes simultanées (*MultiConstrained Problem : MCP)* comme le délai, la gigue, la perte de paquets et le nombre de sauts est un problème beaucoup plus difficile à résoudre que le problème avec une seule contrainte (Cui *et al.*, 2004, Kuipers *et al.*, 2002). Cui *et al.* (2003) et Yuan (2002) proposent de conserver à chaque nœud un chemin « optimal » déterminé à l'avance en fonction des contraintes considérées. Toutefois, cette approche, dans le cadre de connexions dynamiques, peut résulter en l'utilisation de chemins non optimaux si l'état du réseau a changé depuis le dernier calcul de chemin. Pour deux contraintes, Jaffe (1994) suggère de combiner linéairement les métriques reliées à chaque contrainte. Cela permet d'obtenir une métrique composite sur chaque lien pour le routage mais les résultats montrent que la solution obtenue n'est pas toujours faisable. Par ailleurs, l'algorithme de Fallback décrit par Kuipers *et al.* (2002) propose de calculer séquentiellement les plus courts chemins par rapport à une métrique donnée en espérant qu'un des résultats puisse satisfaire toute les contraintes, ce qui n'est pas garanti.

Une autre approche consiste à faire un CAC localement à un lien ou un nœud. Dans ce cas, le lien ou le nœud à un seuil pour chaque paramètre considéré (Cui *et al.*, 2004,

Nordstrom et Dziong, 2006, Spitler et Lee, 2003). La complexité du problème est réduite mais la QdS de bout en bout pourrait ne pas être respectée.

En règle générale, un fournisseur de service doit garantir les paramètres de QdS à toutes les connexions requérant de la QdS, et ce durant toute la durée de la connexion. Comme la plupart des critères de QdS sont liés à la quantité de flots sur les liens, l'impact de l'acceptation d'une nouvelle connexion doit être évalué avant de prendre une décision. Khan *et al.* (2003) ont introduit un modèle d'utilité pour le routage optimal et le CAC. À l'arrivée d'une nouvelle requête, l'objectif est de maximiser une fonction de revenue tout en respectant les contraintes de QdS pour toutes les connexions du réseau. Les auteurs calculent k plus courts chemins réalisables pour chaque connexion du réseau sans tenir compte de la présence des autres connexions. Il s'agit donc, pour chaque requête, de résoudre un problème de routage de l'ensemble des connexions pour trouver la configuration qui maximise les revenus, ce qui requiert un temps important en fonction de la taille du réseau et du nombre de connexions. Par ailleurs, l'optimalité est atteinte en permettant un reroutage de connexions déjà fonctionnelles, ce qui entraîne des interruptions de service.

Ali *et al.* (2005b) considèrent la contrainte de délai et proposent une solution pour réserver une largeur de bande permettant de ne pas altérer les garanties de délai accordées aux autres connexions. Leur solution est basée sur les travaux de Paresh *et al.* (1994) qui définissent le délai théorique maximal subi par une connexion par rapport à la capacité réservée. Cette solution est intéressante mais ne s'applique qu'au cas de CAC local.

Même si beaucoup de travaux ont été effectués dans ce domaine, la plupart des solutions ont pour objectif de satisfaire les contraintes de la connexion qui demande l'accès au réseau. A notre connaissance, aucune solution proposée ne cherche à satisfaire les contraintes de bout en bout de toutes les connexions présentes sur le réseau sans reroutage. Dans cette thèse, nous voulons introduire une méthode permettant de garantir les requis de QdS de bout en bout pour toutes les connexions en service sur le réseau. Une telle solution, avec des temps d'exécution raisonnables, seraient un outil intéressant pour les opérateurs de réseaux.

CHAPITRE III

ARCHITECTURE DE QUALITÉ DE SERVICE POUR ETHERNET ET TISPAN

Comme nous l'avons mentionné précédemment, la QdS dans les réseaux d'accès est un aspect important pour le déploiement futur des NGN. Il est donc important de proposer des architectures et des protocoles pour garantir certains paramètres aux différentes applications. Dans ce chapitre, nous proposons d'abord une architecture de QdS pour les réseaux Ethernet. Dans un deuxième temps, nous intégrons les réseaux Ethernet dans les réseaux d'accès TISPAN en gardant pour objectif de conserver une QdS de bout ne bout. Enfin, ce chapitre se termine par la validation formelle de protocoles conçus pour la réservation de ressources.

3.1 Architecture et protocoles pour un réseau Ethernet

Dans cette section, nous allons définir une architecture et des protocoles pour effectuer le contrôle d'admission et s'assurer que les flots reçoivent le niveau de service requis. Les paramètres de QdS qui nous importent sont le débit, le délai et le niveau de priorité.

3.1.1 Cadre de travail

Nous supposons ce qui suit:

- les nœuds frontières du domaine Ethernet ont des fonctionnalités pour l'inspection des entêtes IP et la classification des paquets avant de les transférer sous forme de trames. Cette hypothèse est censée puisque le réseau Ethernet sera connecté à différents types de technologies d'accès et la tendance est d'utiliser IP de bout en bout ;
- les commutateurs internes implémentent une discipline de service de priorité (PQ: *priority queuing)*. Avec PQ, une file ne peut être servie que si les files de priorités

supérieures sont vides. C'est la discipline de service implémentée par défaut dans les commutateurs supportant la norme 802.1Q ;

- nous choisissons une approche centralisée de CAC pour alléger le fonctionnement des nœuds frontières. En plus, si on garde en idée que l'on voudrait insérer notre réseau Ethernet dans une architecture TISPAN (Figure 2.1), l'approche centralisée est intéressante car c'est la même que TISPAN.

3.1.2 Architecture

La Figure 3.1 présente l'architecture proposée. Elle contient un contrôleur qu'on nommera A-RACF$_{ETH}$ pour se conformer à la nomenclature de TISPAN. Le A-RACF$_{ETH}$ est connecté logiquement à chaque nœud frontière (N_F) par une interface appelée I_{eth}.

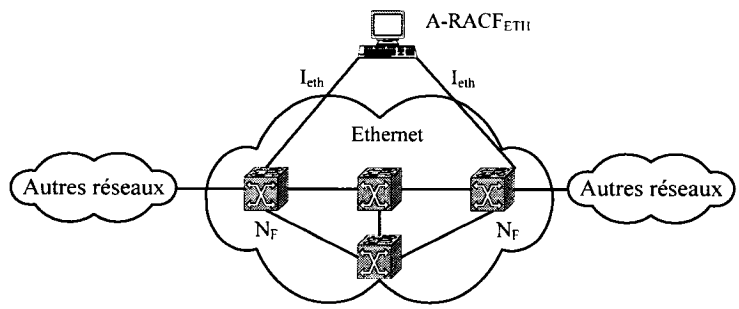

Figure 3.1 Architecture de qualité de service pour Ethernet

3.1.2.1 Rôle du A-RACF$_{ETH}$

Le A-RACF$_{ETH}$ maintient une carte de la topologie physique et une autre de la topologie logique. La topologie logique peut être basée sur les arbres de recouvrement qui sont déployés. En effet, pour éviter les boucles de retransmission, les commutateurs définissent un arbre de recouvrement et bloquent certains ports. Si on décide d'employer une approche basée sur la commutation d'étiquette, la topologie logique représente une carte des différents LSP en fonction sur le réseau Ethernet. En plus de cela, le A-RACF$_{ETH}$ conserve l'utilisation des liens pour chaque classe de service (CdS). Avec ces

informations, le A-RACF$_{ETH}$ peut décider du EVC à emprunter ainsi que de la CdS à utiliser pour un flot donné.

En plus, le A-RACF$_{ETH}$ peut effectuer le contrôle d'admission selon deux modes: réservation ou mesure. Dans le CAC basé sur les réservations (CACR), le A-RACF$_{ETH}$ maintient un état des réservations par classes pour chaque nœud du réseau. À chaque nœud, des seuils sont fixés pour le maximum de largeur de bande qu'il est possible de réserver par classe afin de ne pas pénaliser les trafics de classes inférieures. Le CAC est fait en fonction de l'état des ressources effectivement réservées dans les nœuds. D'autre part, le CAC basée sur les mesures (CACM) utilise une approche différente. Chaque N_F envoie périodiquement une information au A-RACF$_{ETH}$ sur l'utilisation effective des ressources par les EVC et les classes de trafics associées. Le A-RACF$_{ETH}$ prend donc ses décisions en fonction de l'état actuel du réseau. Cela nécessite plus de signalisation. Nous privilégions l'approche CACR car elle permet de donner une garantie plus stricte au trafic.

3.1.2.2 Rôle des nœuds internes (NI)

Les nœuds internes acheminent les trames selon la discipline PQ comme recommandé par le standard 802.1Q. Nous avons défini un mécanisme pour augmenter le nombre de classes de services. Les N_I gardent leur capacité d'apprentissage des adresses MAC ainsi que l'acheminement en fonction du VLAN ID. En effet, il ne faudrait pas altérer le fonctionnement de base des commutateurs Ethernet car certains clients pourraient encore se fier au fonctionnement classique d'Ethernet. Les nœuds internes commutent plus rapidement les trames car ils ne conservent pas d'états relatifs aux flots individuels.

3.1.2.3 Rôle des nœuds frontières (NF)

Les N_F ont un rôle fondamental dans notre architecture. Ils reçoivent les flots IP et conservent les états relatifs aux flots individuels. Ils transfèrent les requêtes vers le A-RACF$_{ETH}$ et les réponses vers les émetteurs. Ce sont eux qui se chargent de vérifier que les flots respectent les contraintes qui leur sont imposées par l'opérateur du réseau

Ethernet notamment sur les paramètres de débit. Si un flot ne respecte pas ces contraintes, les paquets excédentaires sont jetés. Avec ce contrôle de flot aux frontières, il n'est plus nécessaire d'appliquer les mêmes opérations dans les nœuds internes du réseau Ethernet.

Par ailleurs, le N_F est chargé du marquage des trames en fonction de la classe qui a été retournée par le A-$RACF_{ETH}$. Le N_F peut aussi servir de serveur proxy pour effectuer des réservations de ressources au nom de nœuds qui ne supporteraient pas le protocole de réservation de ressources implémenté dans le domaine Ethernet.

3.1.3 Marquage des trames

Les N_F sont responsables de marquer ou de vérifier le marquage des trames. Chaque flot est identifiable par un 5-tuplet. Plusieurs cas peuvent cependant se produire :

- l'application est fournie par une entité installée dans le réseau. Cette entité, en accord avec le A-$RACF_{ETH}$, détermine la CdS qui sera attribuée aux trames du flot concerné permettant ainsi au N_F de marquer les trames.
- le N_S (nœud source) est responsable du marquage des trames. Si le N_S est un élément de confiance, aucune vérification supplémentaire n'est effectuée. Dans le cas contraire, le N_F vérifie la CdS attribuée aux trames.
- aucune information de classification. La classe meilleur effort est utilisée.

Nous supposons que les informations de priorités sont échangées à partir d'un protocole de signalisation qui peut être RSVP ou NSIS par exemple. La concordance entre les types de trafic et les classes de priorité peut être effectuée en se basant sur la proposition de 802.1p (voir tableau 2.1, chapitre 2).

3.1.4 Protocoles pour la QdS dans un réseau Ethernet

Pour améliorer la gestion de la QdS au niveau Ethernet, nous proposons des modifications au format de la trame ainsi que des protocoles pour la gestion des ressources. On appellera N_{FE} le nœud frontière d'entrée du domaine Ethernet et N_{FS} celui de sortie. Pour commencer, nous allons présenter plusieurs solutions concernant l'utilisation du champ Ethertype de l'entête Ethernet.

3.1.4.1 Utilisation de l'Ethertype

L'Ethertype est une information statique de l'entête Ethernet qui permet d'identifier le protocole encapsulé dans la trame Ethernet afin que le module Ethernet puisse transmettre les données au bon protocole. Dans un environnement MetroEthernet, c'est une information qui, sous certaines conditions, peut ne pas être d'une grande utilité et pourrait donc être remplacée par un autre type d'information de contrôle. Pour ce faire, nous supposons que "seul" le protocole IP sera encapsulé dans Ethernet car IP tend à s'imposer comme protocole réseau. On pourrait ajouter une encapsulation IP supplémentaire aux données non IP. Cela ne créerait pas une trop grande surcharge de signalisation car la portion de trafic non IP sera très faible dans les temps à venir puisqu'on parle de plus en plus de réseau tout-IP.

Pour utiliser l'Ethertype à d'autres fins, on définit de nouveaux types d'étiquettes comprenant les mêmes champs que la C-TAG à la différence de l'identificateur de protocoles (TPID). Les nouveaux TPID utilisés indiquent implicitement que le protocole encapsulé est IP et que le champ Ethertype est utilisé à d'autres fins. À partir de cela, il y a deux utilisations spécifiques de l'Ethertype possible :

- L'Ethertype peut être utilisé pour définir d'autres classes et étendre le nombre de VLAN (Figure 3.2). Pratiquement, on se retrouve avec 16 bits de plus. Une proposition est d'utiliser 2 bits supplémentaires pour définir des CdS, ce qui porte à 5 le nombre total de bits pour les CdS (3 bits actuels + 2 bits de l'Ethertype). On définit ainsi un maximum de 32 classes. Il nous reste alors 26 bits pour définir les VLAN (12 bits actuels + 14 bits de l'Ethertype), soit 67 108 864 VLANs possibles. On peut ainsi associer un VLAN ID à chaque EVC sur le réseau Ethernet. Cet espace pourrait aussi être utilisé pour définir des LSP.
- Si on veut conserver la sémantique actuelle des VLAN, on pourrait utiliser les 16 bits de l'Ethertype pour faire de la commutation d'étiquettes à partir de l'entête Ethernet (Figure 3.3). Ces 16 bits pourraient avoir une signification locale pour un lien donné ou globale pour le réseau Ethernet. La commutation ne se ferait plus en fonction de l'adresse MAC destination et des filtres, mais en fonction de cette

étiquette. L'avantage de cela est que l'on conserve les informations essentielles aux commutateurs classiques pour acheminer les trames. Par ailleurs, les 16 bits de l'Ethertype pourraient être utilisés pour transporter le champ TCI d'une autre étiquette VLAN au lieu d'en rajouter une comme dans le standard IEEE 802.1ad.

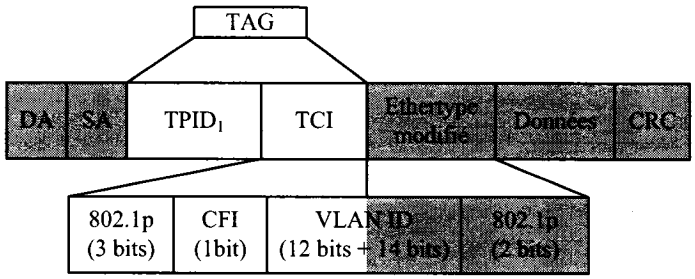

Figure 3.2 Trame Ethernet avec extension du nombre de VLAN et de classes

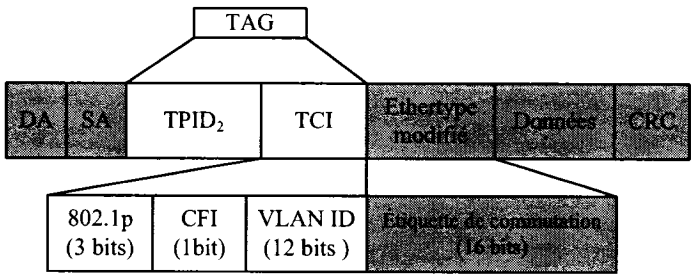

Figure 3.3 Trame Ethernet avec utilisation de l'Ethertype pour la commutation

Un aspect important dans l'utilisation de l'Ethertype est que la taille de l'entête reste la même bien que de nouvelles fonctionnalités soient ajoutées. C'est un avantage considérable pour des applications comme la voix sur IP où le ratio données utiles/longueur du paquet peut être en dessous des 50% dépendamment du codage utilisé.

3.1.4.2 Protocole de réservation de ressources

Après avoir présenté notre architecture de QdS, nous allons maintenant définir les protocoles qui permettront de réserver et de relâcher les ressources dans un réseau Ethernet. Nous appellerons N_S, le nœud source et N_D le nœud destination. Les protocoles que nous allons définir comporte des cas unidirectionnels et des cas bidirectionnels.

Cas unidirectionnel

La Figure 3.4 présente le scénario de réservation pour une session unidirectionnelle de N_S vers N_D. On suppose que N_S connaît les capacités de N_D et que ce dernier acceptera la session si elle est autorisée par le réseau Ethernet. Cela peut être effectué par une signalisation préliminaire.

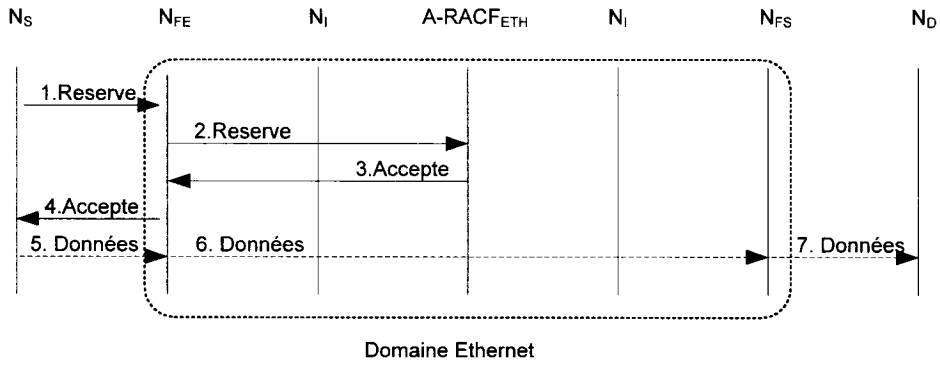

Figure 3.4 Réservation unidirectionnelle de ressources dans un réseau Ethernet

Les étapes de la réservation unidirectionnelle de ressources sont les suivantes:

1. le nœud N_S envoie un message Reserve au N_{FE} pour réserver les ressources. Ce message contient les caractéristiques du flot: type de trafic, débit moyen et débit crête, délai maximal, pertes de paquet maximales, CdS désirée. Ce message peut être bâti en fonction des informations contenues dans un message PATH de RSVP ou un message NSIS qui est aussi un protocole de signalisation de ressources ;
2. le N_{FE} transfère la requête au A-$RACF_{ETH}$. Ce dernier examine la topologie logique associée à la classe de service concernée pour déterminer si le service de cette requête avec les paramètres demandés ne dégraderait pas les performances du réseau. Le A-$RACF_{ETH}$ détermine aussi quel VLAN ID attribuer à ce flot. On suppose que le champ Ethertype est utilisé pour étendre le nombre de VLAN ID ;
3. le A-$RACF_{ETH}$ accepte la requête. Il retourne au N_{FE} la confirmation concernant la CdS, le VLAN ID et l'arbre de recouvrement à utiliser pour acheminer le trafic du flot concerné ;
4. le N_{FE} bâtit l'état de réservation et informe N_S de l'acceptation de la requête ;

5. le N_S commence à émettre des données ;
6. le N_{FE} forme les trames Ethernet après avoir inspecté l'entête IP. Il rajoute une étiquette VLAN et formate le champ Ethertype pour inclure la CdS et le VLAN ID associé avant de transférer les trames. Le N_{FE} se charge aussi de rejeter le trafic si les paramètres ne sont pas respectés ;
7. le N_{FS} retire l'entête Ethernet. On suppose que le N_{FS} se base sur l'entête IP pour acheminer les données au N_D.

La Figure 3.5 présente le processus de libération unidirectionnelle de ressources.

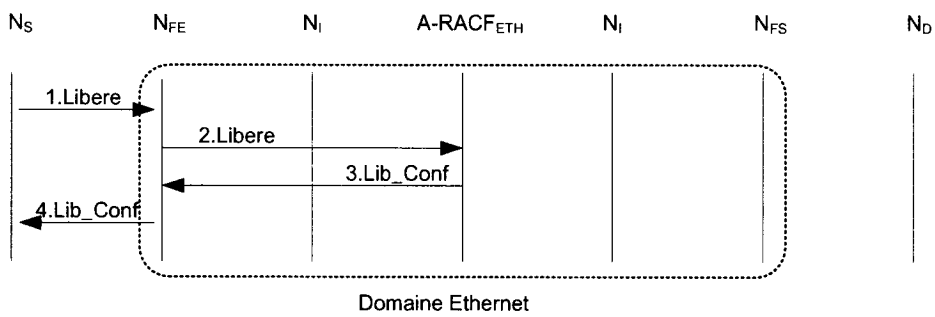

Figure 3.5 Libération unidirectionnelle de ressources dans un réseau Ethernet

Les étapes de la libération de ressources sont les suivantes:

1. le nœud N_S envoie un message Libere au N_{FE} pour libérer les ressources. Ce message contient les identifiants du flot et peut être bâti en fonction des informations contenues dans un message PATHTEAR de RSVP ou un message NSIS ;
2. le N_{FE} détruit les états liés au flot concerné. Il met à jour les informations concernant la quantité de trafic associée à la classe et au port concerné. Il transfère ensuite la requête Libere au $A\text{-}RACF_{ETH}$;
3. le $A\text{-}RACF_{ETH}$ met à jour les données concernant la topologie logique associée à la classe considéré. Il confirme au N_{FE} le relâchement des ressources par un message Lib_Conf ;

4. le N_{FE} fait suivre le message Lib_Conf au N_S pour lui signifier la libération effective des ressources.

Cas bidirectionnel

La Figure 3.6 présente respectivement le scénario de réservation pour une session bidirectionnelle de N_S vers N_D. On suppose que N_S connaît les capacités de N_D et que ce dernier acceptera la session si elle est autorisée par le réseau Ethernet.

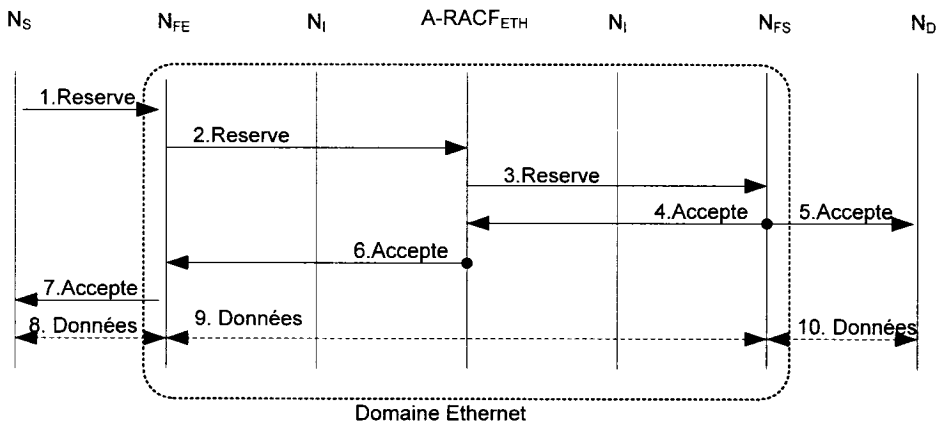

Figure 3.6 Réservation bidirectionnelle de ressources dans un réseau Ethernet

Les étapes de la réservation bidirectionnelle de ressources sont les suivantes:

1. le nœud N_S envoie un message Reserve au N_{FE} pour réserver les ressources. Ce message contient les caractéristiques du flot: type de trafic, débit moyen et débit crête, délai, perte de paquet, CdS désirée. Il contient les paramètres requis pour les deux sens de la communication ;
2. le N_{FE} transfère la requête au A-$RACF_{ETH}$. Ce dernier examine la topologie logique associée à la classe de service concernée pour déterminer si le service de cette requête avec les paramètres demandés ne dégraderait pas les performances du réseau. Le A-$RACF_{ETH}$ détermine quel VLAN ID attribuer à ces flots ;
3. le A-$RACF_{ETH}$ transfère la requête vers le N_{FS} afin que celui-ci effectue les configurations nécessaires ;

4. le N_{FS} bâti l'état de réservation associé au flot répond au A-$RACF_{ETH}$ pour lui indiquer que les ressources ont été réservées ;
5. le N_{FS} informe le N_D de l'acceptation de la requête. On supposera qu'il y a eu un échange préliminaire entre le N_S et le N_D et que le N_D attend cette confirmation ;
6. le A-$RACF_{ETH}$ accepte la requête. Il retourne au N_{FE} la confirmation concernant la CdS, le VLAN ID et l'arbre de recouvrement à utiliser pour acheminer le trafic du flot concerné. ;
7. le N_{FE} bâtit l'état de réservation associé au flot et informe N_S de l'acceptation de la requête ;
8. 9. 10. le N_S et le N_D commencent à émettre des données. Le N_{FE} (N_{FS}) forme les trames Ethernet après avoir inspecté l'entête IP du flot émis par le N_S (N_D). Il rajoute une étiquette VLAN et formate le champ Ethertype pour inclure la CdS et le VLAN ID associé avant de transférer les trames. Le N_{FE} (N_{FS}) se charge aussi de rejeter le trafic de N_S ($N_{D)}$ si les paramètres ne sont pas respectés. Le N_{FS} (N_{FE}) retire l'entête Ethernet des trames associées au flot provenant de N_S (N_D). On suppose que le N_{FS} (N_{FE}) se base sur l'entête IP pour acheminer les données au N_D (N_S).

La Figure 3.7 présente le processus de libération bidirectionnelle de ressources. Les étapes de la libération de ressources sont les suivantes:

1. le nœud N_S envoie un message Libere au N_{FE} pour libérer les ressources. Ce message contient les identifiants du flot et peut être bâti en fonction des informations contenues dans un message PATHTEAR de RSVP ou un message NSIS ;
2. le N_{FE} détruit les états liés au flot concerné. Il met à jour les informations concernant la quantité de trafic associée à la classe et au port concerné. Il transfère ensuite la requête Libere au A-$RACF_{ETH}$;
3. le A-$RACF_{ETH}$ met à jour les données concernant la topologie logique associée à la classe considérée. Il transfère ensuite la requête Libere au N_{FS} ;

4. le N_{FS} détruit les états liés au flot concerné. Il met à jour les informations concernant la quantité de trafic associée à la classe et au port concerné. Il confirme au A-$RACF_{ETH}$ le relâchement des ressources par un message Lib_Conf. Simultanément, il informe le N_D du fait que les ressources ne sont plus actives par le message Lib_Conf ;
5. le A-$RACF_{ETH}$ transfère le message Lib_Conf au N_{FE} pour lui signifier la libération des ressources ;
6. le N_{FE} fait suivre le message Lib_Conf au N_S pour lui signifier la libération effective des ressources.

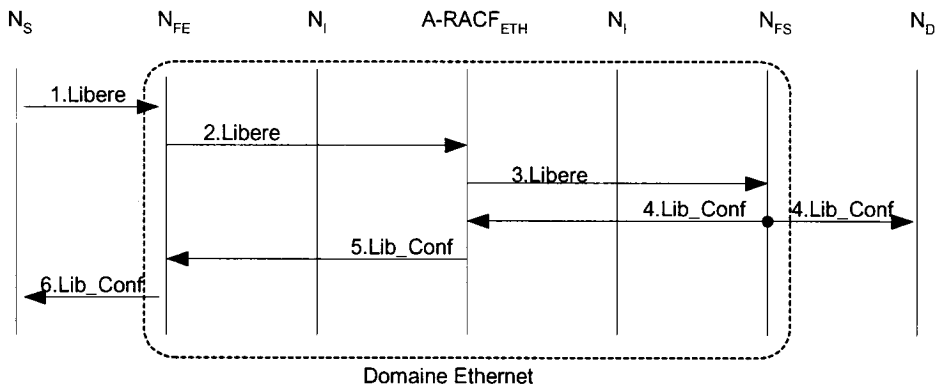

Figure 3.7 Libération bidirectionnelle de ressources dans un réseau Ethernet

3.2 Intégration d'Ethernet au réseau d'accès TISPAN

Dans la section précédente, nous avons proposé une architecture et des protocoles permettant de réserver des ressources dans un réseau Ethernet pour un meilleur service des utilisateurs. Comme nous l'avions mentionné précédemment, nous visons l'intégration d'un réseau Ethernet dans un réseau d'accès TISPAN. Cette section présente une architecture et des protocoles permettant une telle intégration.

3.2.1 Architecture proposée

La Figure 3.8 montre l'architecture physique proposée tandis que la Figure 3.9 présente une architecture fonctionnelle.

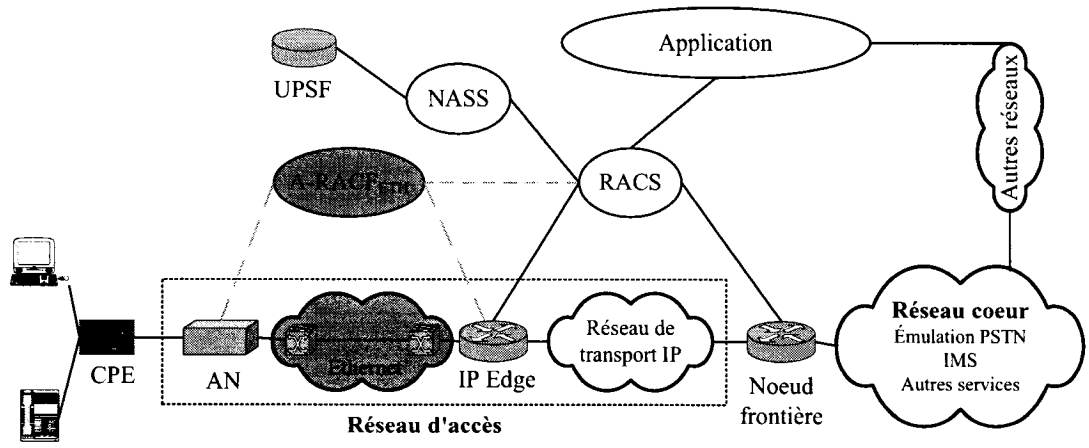

Figure 3.8 Architecture de réseau d'accès TISPAN incluant Ethernet

Le réseau Ethernet est intégré comme réseau d'agrégation L2. Nous considérons que le contrôle du réseau Ethernet est indépendant de celui du réseau TISPAN qui est effectué par le RACS. Ce choix a été fait pour permettre l'utilisation de réseaux appartenant à différents opérateurs.

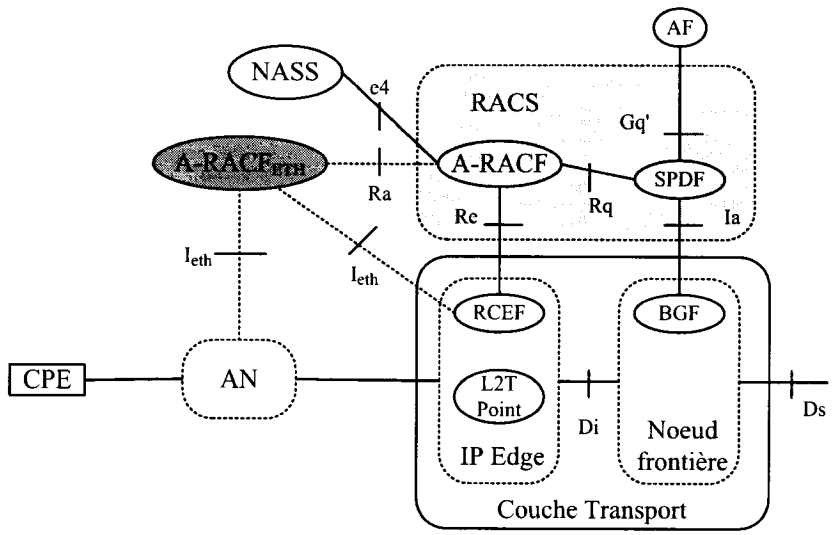

Figure 3.9 Architecture fonctionnelle proposée pour la QdS

Le RACS est connecté au A-RACF$_{ETH}$ par l'interface R_a. Le A-RACF$_{ETH}$ est connecté au IP Edge et à l'AN qui font office de nœuds frontières pour le réseau Ethernet. Ce sont eux qui vérifieront que les flots respectent les paramètres négociés. Ils

marquent aussi les trames avec les informations sur les VLAN ID et la classe de service. Plus spécifiquement, on voit sur la Figure 3.9 que le A-RACF$_{ETH}$ interagit directement avec le RCEF. Il sera donc nécessaire d'étendre les fonctionnalités du RCEF pour supporter l'interaction avec le A-RACF$_{ETH}$. D'autre part, la décision d'admettre ou de rejeter une requête incombe au RACS. Le A-RACF$_{ETH}$ transmet des informations sur les capacités du réseau Ethernet à desservir la connexion. Ces informations sont prises en compte par le RACS pour la décision finale. Ce choix est fait car le RACS a une vue de la capacité globale du réseau.

3.2.2 Protocole de réservation de ressources

Dans cette section, nous présenterons les procédures de réservation et de libération de ressources. Nous présentons à chaque fois le processus actuel utilisé dans TISPAN ainsi que le processus proposé. Les étapes additionnelles sont indiquées en gras. AF (*Application Function*) désigne un serveur d'application. Nous supposons que le SPDF accepte la requête et l'interaction avec le BGF n'est pas présentée car l'action du SPDF est peu importante pour notre recherche. Comme TISPAN, les figures présentées utilisent le modèle *QoS Push* où le AF demande l'autorisation et la réservation au RACS.

3.2.2.1 Réservation de ressources

Processus actuel (Figure 3.10)

1. AF initie une réservation de ressources pour des flots donnés. Il envoie une requête de service vers le SPDF.
2. Le SPDF vérifie la requête par rapport aux politiques de l'opérateur pour l'AF considéré. Il l'autorise et envoie une réservation de ressources vers le A-RACF (Ressource_Req).
3. Le A-RACF effectue le contrôle d'admission et l'autorisation en se basant sur les politiques locales du réseau d'accès. Il décide de la nécessité de faire appliquer des politiques de trafic par le RCEF.
4. Le A-RACF demande au RCEF d'appliquer les politiques de trafic.

5. Le RCEF confirme l'installation des politiques et la réservation de ressources au A-RACF (Ressource_Cnf).
6. Le A-RACF confirme la réservation au SPDF.
7. Le SPDF confirme la réservation au AF.

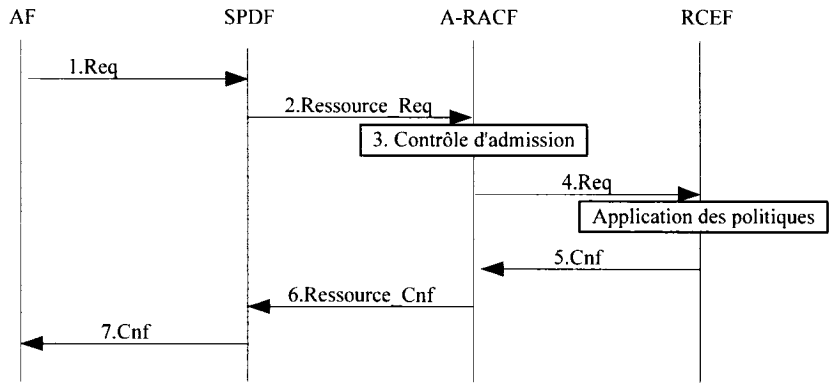

Figure 3.10 Réservation de ressources actuelle pour le RACS de TISPAN

Processus proposé (Figure 3.11)

1. AF initie une réservation de ressources pour des flots donnés. Il envoie une requête de service vers le SPDF.
2. Le SPDF vérifie la requête par rapport aux politiques de l'opérateur pour l'AF considéré. Il l'autorise et envoie une réservation de ressources vers le A-RACF (Resource_Req).
3. Le A-RACF effectue le contrôle d'admission et l'autorisation en se basant sur les politiques locales du réseau d'accès. Il décide de la nécessité de faire appliquer des politiques de trafic par le RCEF.
4. **Le A-RACF achemine la requête de QdS au A-RACF$_{ETH}$ qui vérifie que le flot en question peut être autorisé. Si oui, il effectue les réservations de ressources nécessaires selon le modèle de QdS implémentée dans le réseau Ethernet.**
5. **Le A-RACF$_{ETH}$ confirme la réservation au A-RACF.**
6. Le A-RACF demande au RCEF d'appliquer les politiques de trafic.
7. Le RCEF confirme l'installation des politiques et la réservation de ressources au A-RACF (Ressource_Cnf).

8. Le A-RACF confirme la réservation au SPDF.
9. Le SPDF confirme la réservation au AF.

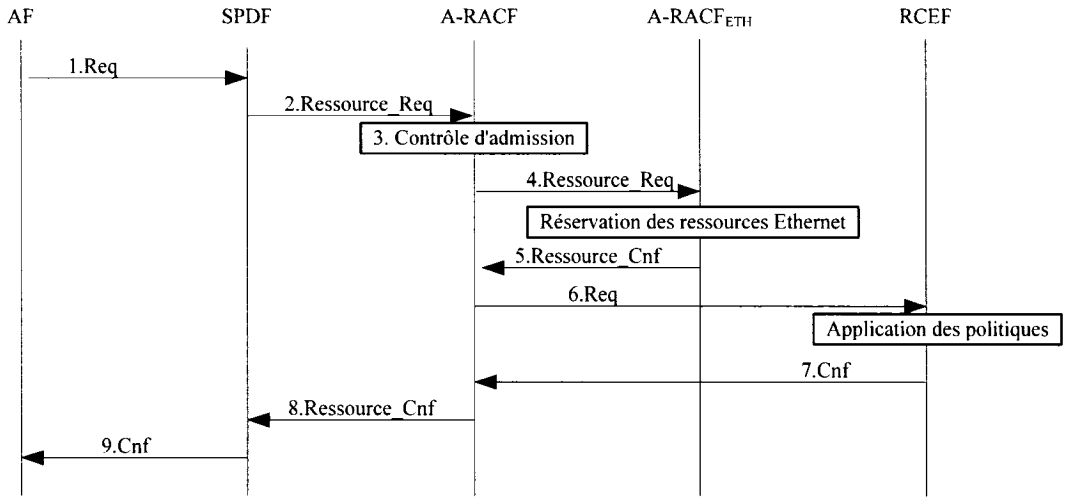

Figure 3.11 Réservation de ressources proposée pour le RACS de TISPAN

3.2.2.2 Libération de ressources

Processus actuel (Figure 3.12)

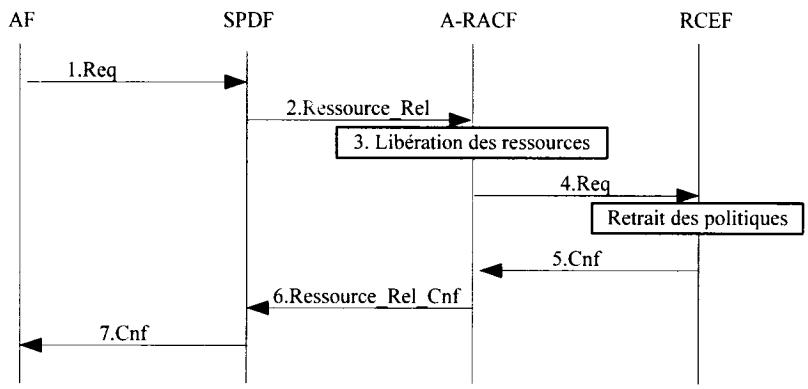

Figure 3.12 Libération de ressources actuelle pour le RACS de TISPAN

1. AF envoie une requête de libération de ressources vers le SPDF.
2. Le SPDF envoie une requête de libération de ressources vers le A-RACF (Ressource_Rel).
3. Le A-RACF libère les ressources associées et évalue la nécessité de demander au RCEF de retirer certaines politiques de trafic mises en place.

4. Le A-RACF demande au RCEF de retirer les politiques de trafic pour les flots concernés.
5. Le RCEF confirme le retrait des politiques au A-RACF.
6. Le A-RACF confirme la suppression des politiques au SPDF (Ressource_Rel_Cnf).
7. Le SPDF confirme la libération des ressources au AF.

Processus proposé (Figure 3.13)

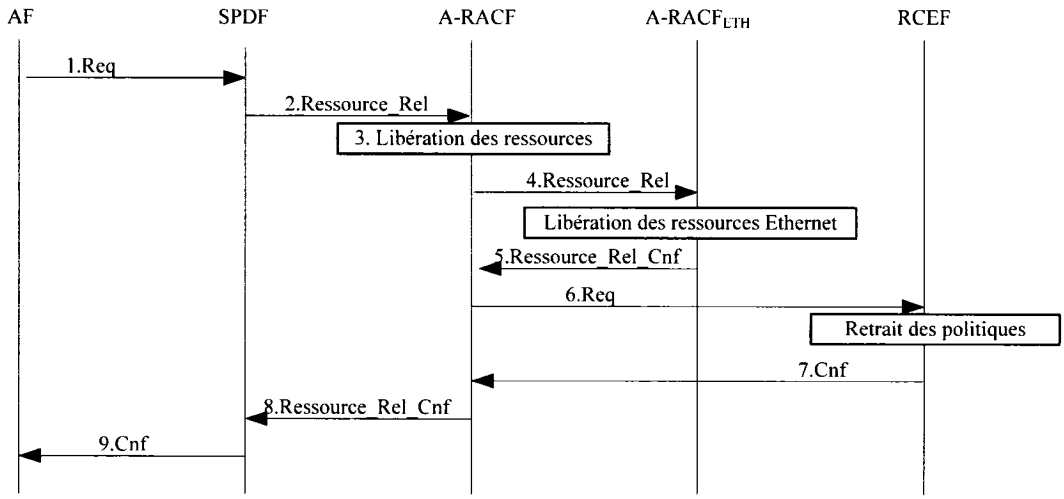

Figure 3.13 Libération de ressources proposée pour le RACS de TISPAN

1. AF envoie une requête de libération de ressources vers le SPDF.
2. Le SPDF envoie une requête de libération de ressources vers le A-RACF (Resource_Rel).
3. **Le A-RACF achemine la requête de libération de ressource au A-RACF$_{ETH}$ qui effectue le relâchement des ressources dans le réseau Ethernet.**
4. **Le A-RACF$_{ETH}$ confirme la libération au A-RACF.**
5. Le A-RACF libère les ressources associées et évalue la nécessité de demander au RCEF de retirer certaines politiques de trafic mises en place.
6. Le A-RACF demande au RCEF de retirer les politiques de trafic pour les flots concernés.
7. Le RCEF confirme le retrait des politiques au A-RACF.

8. Le A-RACF confirme la suppression des politiques au SPDF (Resource_Rel-Cnf).
9. Le SPDF confirme la libération des ressources au AF.

3.2.2.3 Contenu des messages

Lorsque le A-RACF envoie le message *Ressource_Req* au A-RACF$_{ETH}$, il spécifie plusieurs paramètres qui sont: le nœud d'accès AN, le nœud IP Edge concerné, le type de trafic, un identifiant de flot, le débit moyen et crête, le taux de perte, la gigue et le délai maximal tolérés par l'application. Le A-RACF$_{ETH}$ répond en spécifiant qu'il accepte le flot concerné. Il est possible aussi qu'il y ait une négociation entre le A-RACF$_{ETH}$ et le A-RACF au sujet de certains paramètres. Lors de la libération de ressources par le message *Ressource_Rel*, le A-RACF indique au A-RACF$_{ETH}$ quelle réservation détruire en lui spécifiant l'identifiant du flot.

Suite à la présentation de nos solutions, nous allons valider formellement le protocole.

3.3 Validation formelle

La validation formelle consiste à vérifier que le protocole se conforme à certaines propriétés. C'est une méthode d'évaluation qui, sans donner d'éléments sur la performance du protocole, permet néanmoins de juger de façon logique le fonctionnement global du protocole. La validation comporte les 3 étapes suivantes :

- modélisation du système ;
- spécification de propriétés attendues du système ;
- preuve que le modèle répond bien aux propriétés spécifiées.

Pour notre validation, nous utilisons le logiciel UPPAAL qui est développé depuis 1995 par l'université d'Uppsala. UPPAAL est implémenté en langage C++ et est composé de 3 modules qui sont :

- une interface graphique pour la modélisation du système ;
- un simulateur graphique du système à modéliser. Ce simulateur permet de suivre en temps réel la séquence d'exécution des étapes du modèle afin de vérifier s'il correspond au protocole ;

• un vérificateur des propriétés du modèle (*Model checker*).

Avec UPPAAL, la modélisation des processus se fait sous formes d'automates temporisés composés d'un ensemble d'états et de transitions. Les transitions entre états d'un même processus peuvent être directes ou conditionnelles et impliquer des mises à jour de certaines variables. Chaque entité de notre protocole est représentée par un automate et ces automates doivent être synchronisés pour décrire adéquatement le fonctionnement du protocole. Lors d'une transition, un automate donné peut déclencher une opération dans *n* autres automates en lançant une synchronisation particulière. Par ailleurs, l'ensemble des automates constitue le système étudié. Ensuite, il faut définir les propriétés que l'on veut étudier à l'aide de la logique temporelle CTL. Enfin, nous validons les propriétés d'accessibilité des états et de non blocage par le *model checker*.

3.3.1 Algorithmes

Nous allons maintenant présenter les algorithmes utilisés par les différents éléments du système. Pour chaque algorithme, les rectangles représentent des états du système tandis que les parallélogrammes indiquent une décision à prendre en fonction des réponses obtenus suite aux requêtes formulées.

3.3.1.1 AF

Le serveur d'application est une entité située dans le réseau. Elle peut appartenir au réseau du fournisseur d'accès comme être une tierce entité dans un réseau distant. La Figure 3.14 illustre l'algorithme utilisé pour représenter le fonctionnement du serveur d'applications. A partir de l'état d'attente, l'application a le choix de faire une demande de réservation de ressources dans la Figure 3.14. Si la demande est effectuée, l'application envoie la requête vers le SPDF, sinon elle reste dans l'état d'attente. Nous n'avons mentionné que les états liés au processus de réservation de ressources. Si la requête est acceptée par tous, AF passe dans un état de confirmation avant la fin de la tentative. Dans le cas contraire, l'état rejet est visité avant de faire une nouvelle tentative au besoin.

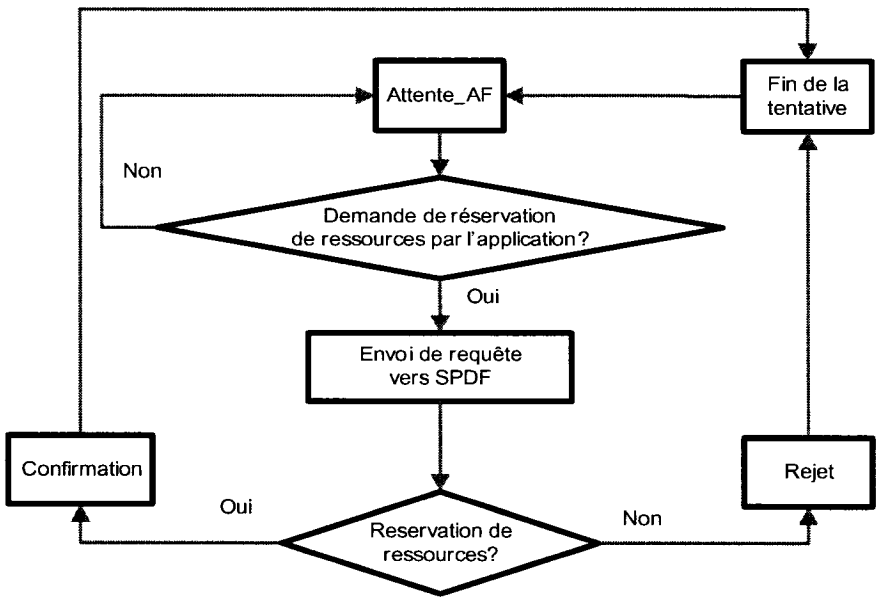

Figure 3.14 Algorithme de réservation au niveau du AF

3.3.1.2 SPDF

Lorsqu'une requête de réservation est reçue par le SPDF qui est dans l'état d'attente, celui-ci fait une vérification de politique avant d'autoriser la poursuite de la réservation. Si l'autorisation n'est pas accordée par le SPDF, celui-ci passe dans l'état rejet avant d'informer le AF par un message. Si le SPDF autorise la connexion, il faut envoyer la requête au A-RACF. Si la réservation est rejetée par un élément autre que le SPDF, un message de rejet est envoyé au AF sinon une confirmation de réservation est transmise au AF. Lorsque la session est en cours, s'il y a demande de fin de session par le AF, on passe à l'état libération de ressources. Ce dernier état est composé de plusieurs étapes mais, pour simplification, il est représenté par un seul rectangle à la Figure 3.15.

3.3.1.3 A-RACF

L'algorithme du A-RACF présenté à la Figure 3.16 est le plus complexe. Quand une requête de réservation est reçue par le A-RACF qui est dans l'état d'attente, celui-ci vérifie que les ressources sont disponibles dans le réseau de transport IP qui est sous son contrôle direct. S'il n'y pas assez de ressources disponibles au niveau IP, la connexion est rejetée et le SPDF est informé. D'autre part, si les ressources IP sont suffisantes, une

requête est envoyée vers le A-RACF$_{ETH}$ pour la réservation de ressources Ethernet. Dans la suite de l'algorithme, si la réservation Ethernet est positive, une demande d'application de politique est envoyée au RCEF avant la confirmation de la réservation au SPDF. S'il arrivait que le RCEF ne puisse pas appliquer les politiques, le A-RACF devra d'abord demander une libération de ressources Ethernet avant d'informer le SPDF. Une fois la session en cours, la suite de l'algorithme est identique à celui du SPDF.

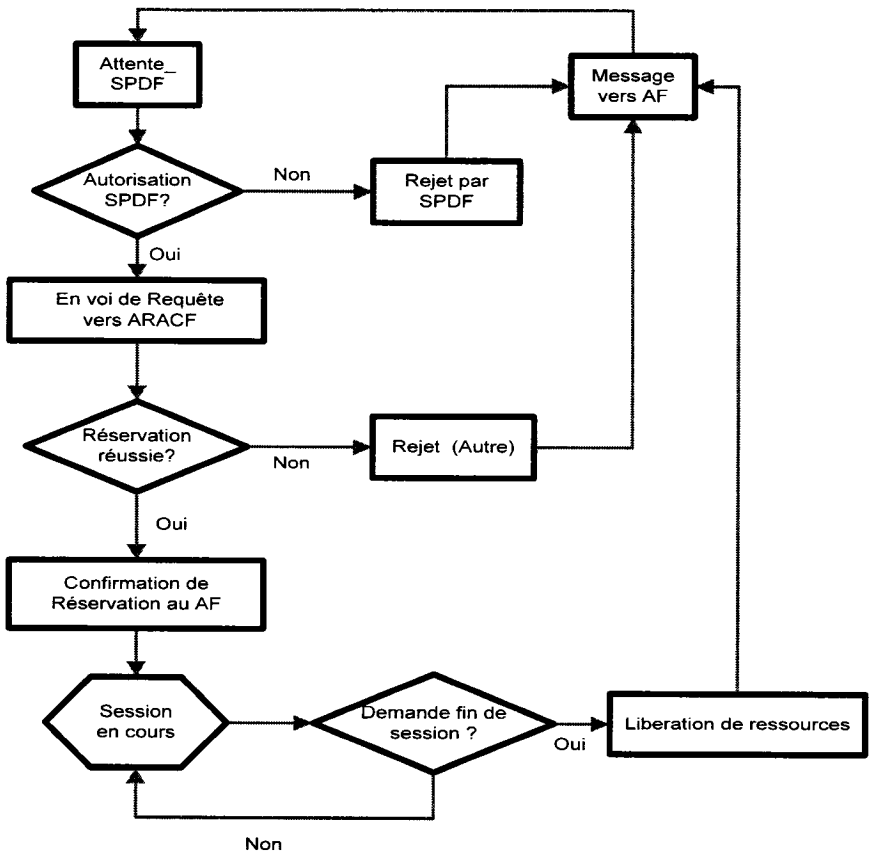

Figure 3.15 Algorithme de réservation au niveau du SPDF

3.3.1.4 A-RACF$_{ETH}$

Pour ce contrôleur, l'algorithme est présenté à la Figure 3.17. À la réception d'une requête de réservation par le A-RACF$_{ETH}$ qui est dans l'état d'attente, celui-ci vérifie que les ressources sont disponibles sur le réseau Ethernet et les réserve. Si la réservation est

impossible, le A-RACF$_{ETH}$ informe le A-RACF. Si la réservation est effectuée, une confirmation est envoyée vers le A-RACF. Une fois la session en cours, la suite de l'algorithme est identique à celui du SPDF.

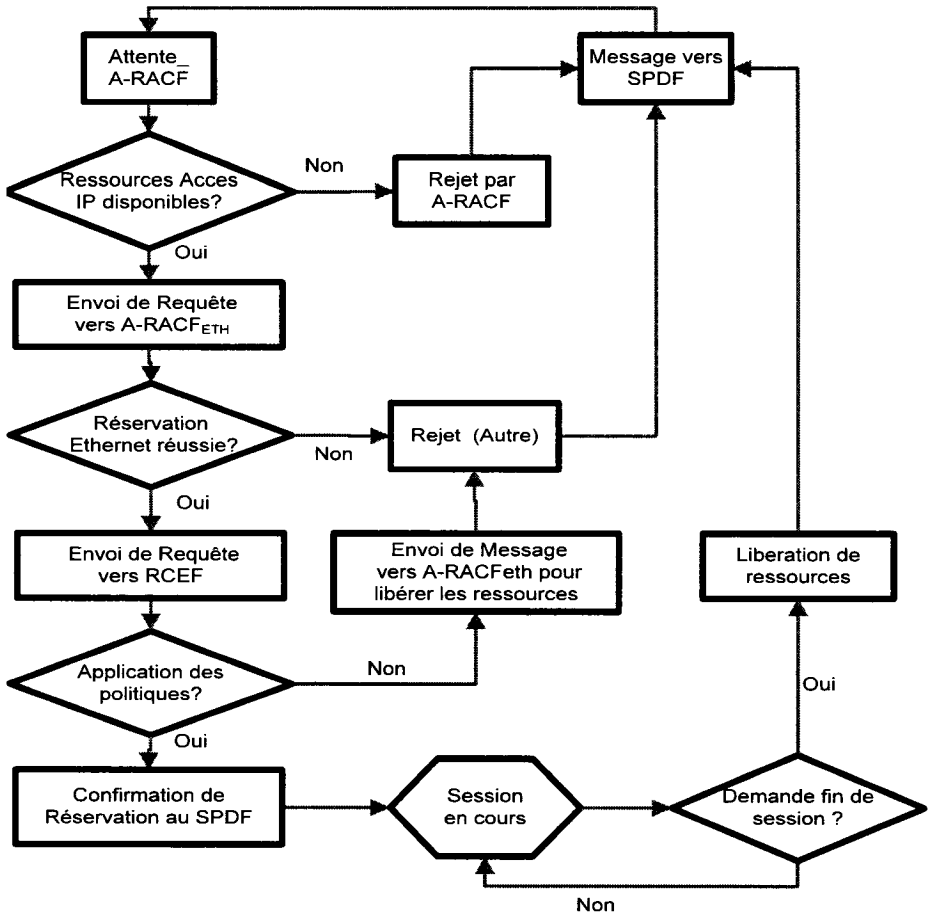

Figure 3.16 Algorithme de réservation au niveau du A-RACF

3.3.1.5 RCEF

La Figure 3.18 illustre l'algorithme du RCEF. Si le RCEF, qui est dans l'état d'attente, reçoit une requête de connexion, les politiques sont appliquées. Si le résultat de cette action est négatif, il faut informer le A-RACF du rejet de la requête. Dans le cas contraire, on confirme au A-RACF l'application des politiques. Une fois la session en cours, la suite de l'algorithme est identique à celui du SPDF.

Les automates que nous avons utilisés sont bâtis à partir des algorithmes présentés. L'annexe A renferme les schémas UPPAAL des différents automates.

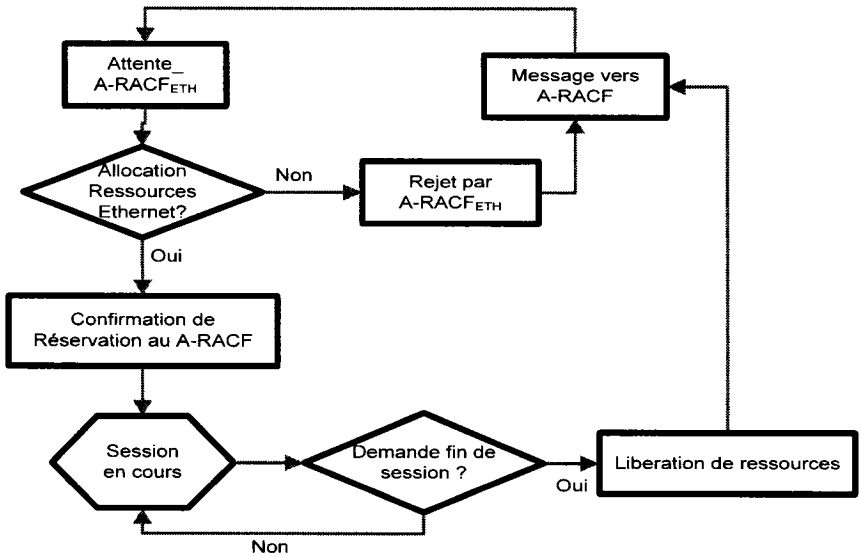

Figure 3.17 Algorithme de réservation au niveau du A-RACF$_{ETH}$

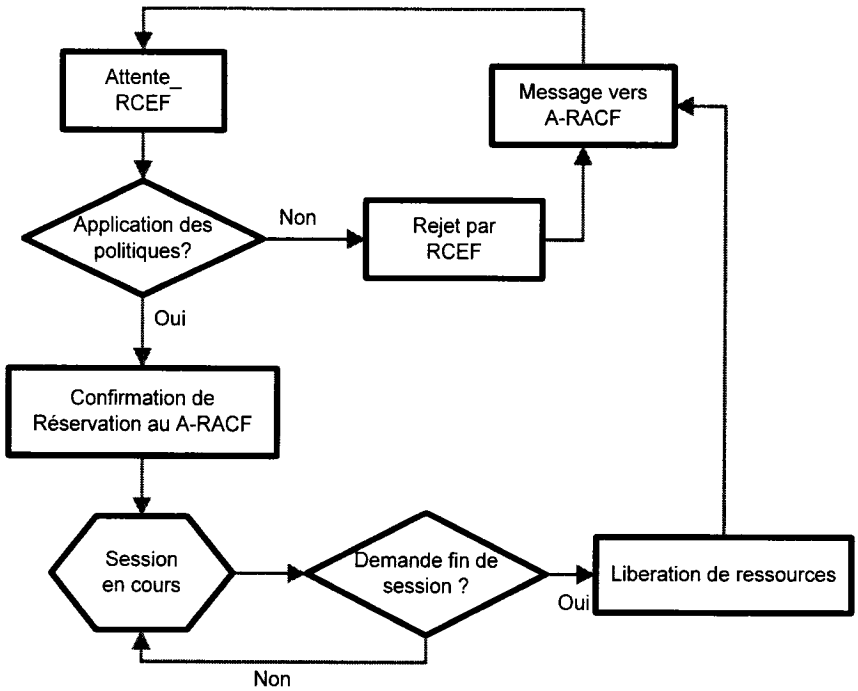

Figure 3.18 Algorithme de réservation au niveau du RCEF

3.3.2 Simulations

Nous avons effectué des simulations pour vérifier que les résultats sont conformes aux spécifications du protocole. La Figure 3.19 traite du cas où la réservation est réussie. Nous pouvons constater que la simulation respecte les différents échanges de messages du protocole de éservation de ressources tel que présenté à la Figure 3.11. La Figure 3.20 présente un cas de rejet d'une requête de connexion. Nous illustrons une situation où le A-RACF refuse la demande. Ce dernier inorme donc le SPDF de sa décision. Enfin, la libération de ressources initiée par le AF est présentée à la Figure 3.21. La succession d'échange de messages obtenue est conforme au processus décrit à la Figure 3.13.

3.3.3 Vérification des propriétés

Nous nous intéressons aux propriétés d'accessibilité des états et de non blocage. En logique CTL, l'accessibilité s'écrit AG (p1 → EF p2) où p1 et p2 sont des propositions. Le non blocage s'écrit AG(EX *true*) où *true* est un état initial quelconque. Avec UPPAAL, on note « E<>Processus.État » pour l'accessibilité d'un état et « A[] not deadlock » pour le non blocage. La vérification a démontré l'absence d'état bloquant avec une accessibilité de tous les états du système.

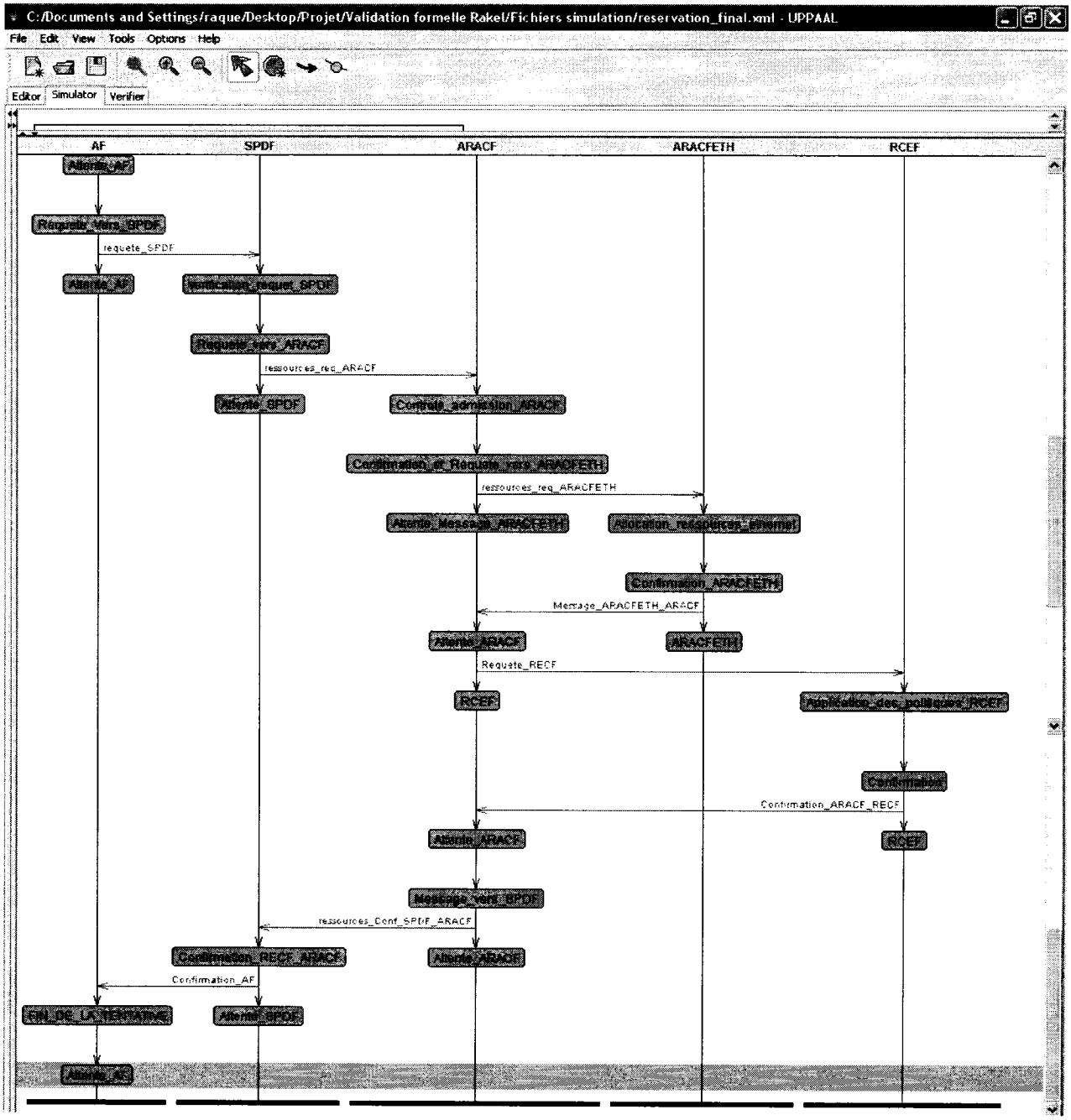

Figure 3.19 Réservation réussie de ressources

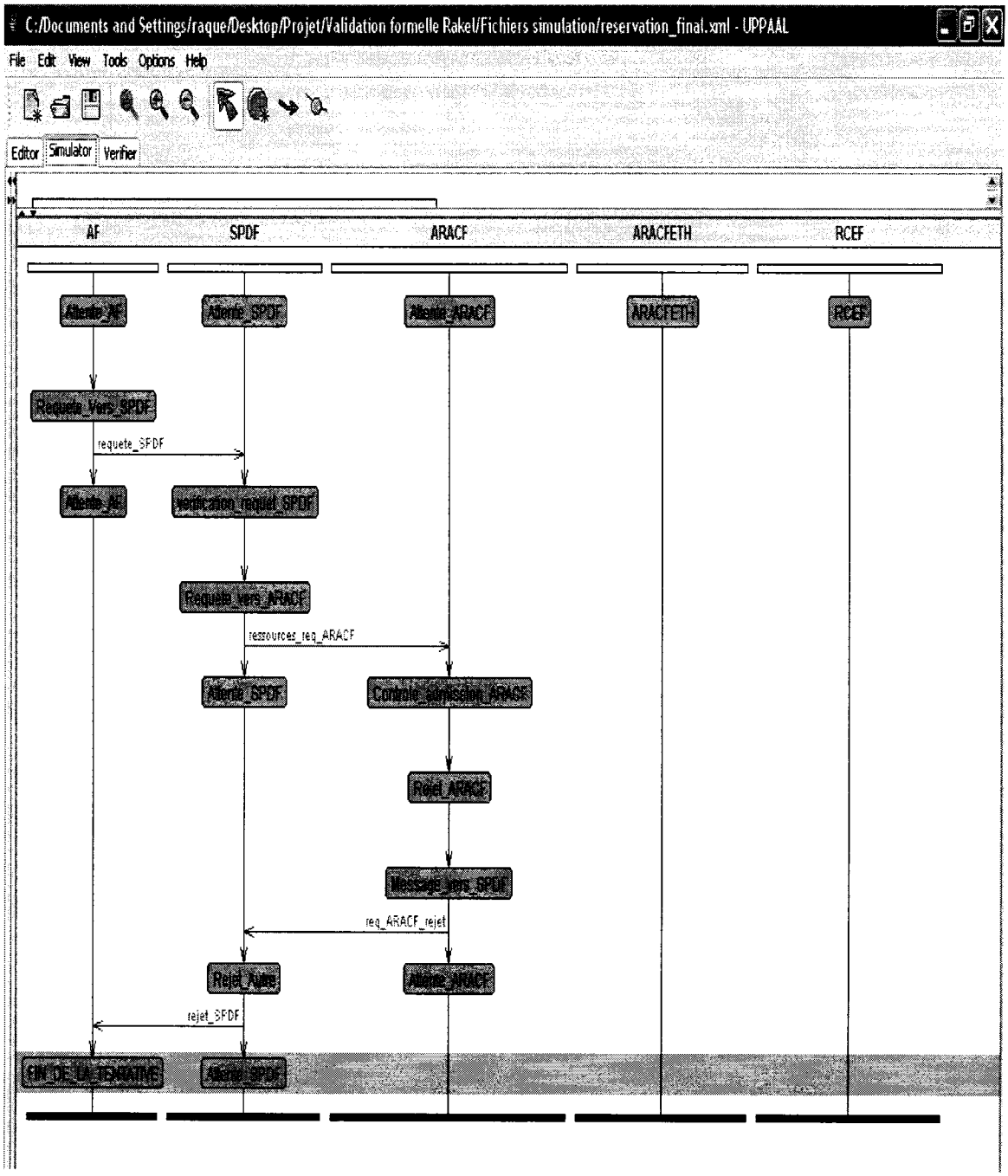

Figure 3.20 Réservation rejetée par le A-RACF

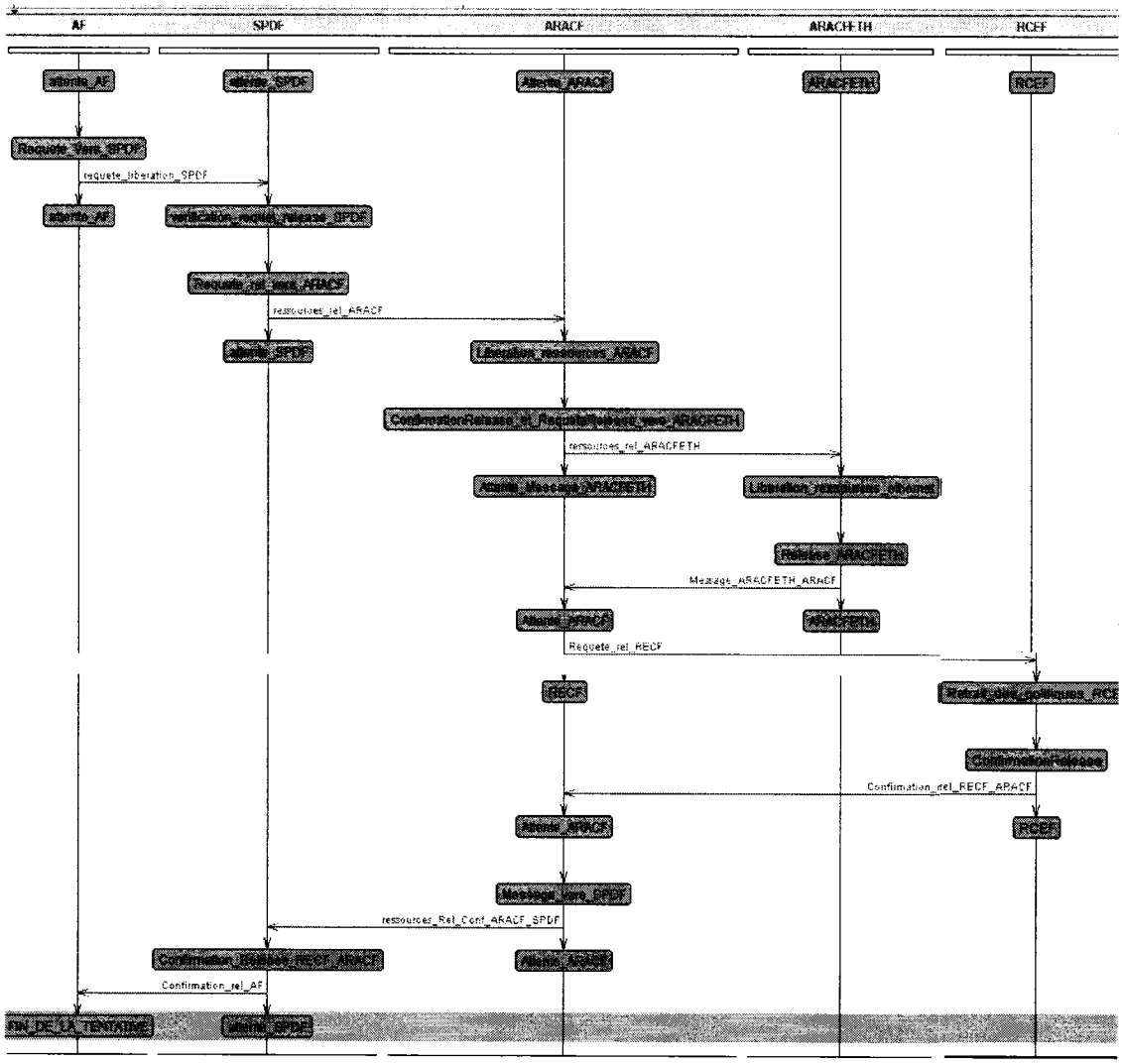

Figure 3.21 Libération de ressources

CHAPITRE IV

ROUTAGE ET CONTRÔLE D'ADMISSION DES CONNEXIONS AVEC CONTRAINTES DE DÉLAI DE BOUT EN BOUT

Le routage et le CAC étant des aspects déterminants pour garantir la QdS, nous proposons dans ce chapitre un modèle de programmation linéaire permettant d'obtenir une solution de routage optimale, et ce, dans des temps relativement courts. Nous comparons notre méthode à certaines propositions de la littérature. Ce chapitre traite du problème de délai de bout en bout. Nous formulerons d'abord un modèle général pour le problème du CAC avec contraintes de délai. Puis, nous présenterons un algorithme et un modèle mathématique linéaire qui nous servira à la résolution du problème considéré. Enfin, nous clôturerons ce chapitre par une évaluation de performance.

4.1 Modélisation mathématique du problème

Le problème du routage de QdS et CAC est souvent représenté par un modèle de programmation linéaire dont l'objectif est soit de minimiser une fonction de coûts ou de maximiser une fonction de revenus. Dans notre cas, nous considérons une fonction de coûts. Généralement, le modèle prend une forme similaire à celle de G :

(G) Minimiser $U\,\mathbf{x}$

sujet à

Contraintes de capacités

Contraintes de QdS

Contraintes diverses

où U est le vecteur de coûts associés aux liens du réseau et $\mathbf{x}$, le vecteur de chemins pour indiquer l'utilisation d'un lien par un flot donné ou pour désigner la quantité de flot sur le

lien. Les contraintes de capacités servent à exprimer les contraintes physiques tandis que les contraintes de QdS (délai, perte de paquets, gigue...) permettent d'offrir un service adapté au besoin du client.

4.1.1 Cadre de travail

Dans la suite de ce chapitre, nous considérerons un réseau physique sur lequel on retrouve une topologie logique constituée de LSP. Le réseau comporte $n = /N/$ nœuds où N est l'ensemble des nœuds et $m = /M/$ liens où M est l'ensemble des liens. Parmi les n nœuds, il y a $n_f = /N_f/$ nœuds frontières (où N_f inclus dans N est l'ensemble des nœuds frontières) qui sont les extrémités des LSP ($n_f \leq n$). Nous supposons que la topologie logique est connue et qu'il existe un LSP entre chaque paire de nœuds frontières. Nous considérons une approche centralisée où un gestionnaire de ressources est chargé de faire le contrôle d'admission. L'objectif est de router les connexions sur cette topologie logique. Nous considérons une seule classe de service.

Les contraintes de QdS étudiées sont le délai et la perte de paquets de bout en bout. Le coût d'un LSP est défini par la somme ou le produit des coûts des différents liens qui composent le LSP selon que la métrique est additive ou multiplicative. Nous considérons trois types d'objectif : minimiser le délai, minimiser la perte de paquets ou minimiser un coût fixe calculé en fonction des coûts fixes des liens. Les modèles que nous proposons permettent de trouver une solution de routage pour une connexion à la fois.

Puisque nos solutions considèrent la satisfaction de la globalité des connexions, nous allons les comparer à des modèles qui ne considèrent que les paramètres de QdS de la connexion qui demande l'accès. Rappelons que notre objectif est de fournir une méthode rapide et efficace car nous faisons un contrôle d'admission dynamique pour chaque connexion. Nous parlerons de nouvelle connexion pour désigner la connexion pour laquelle le CAC est effectuée et de connexions admises pour celles qui sont déjà en fonction sur le réseau. Par ailleurs, au niveau des délais calculés, nous négligeons le délai de traitement et le délai total comprend le temps d'attente, le temps de transmission et le temps de propagation.

Après avoir présenté le cadre de travail, nous allons maintenant définir les ensembles, les variables et les constantes qui seront utilisés pour notre modélisation.

4.1.2 Ensembles, variables et constantes

Voici les ensembles que nous utiliserons pour les différentes modélisations :

- N, l'ensemble des nœuds ;
- $N_f \subset N$, l'ensemble des nœuds frontières ;
- M, l'ensemble des liens unidirectionnels du réseau ;
- L, l'ensemble des LSP unidirectionnels formant la topologie logique ;
- T, l'ensemble des connexions établies sur le réseau. Chaque connexion commence à O(t) et finit à D(t) qui appartiennent à l'ensemble N ;
- C, l'ensemble des chemins constitués d'un ou plusieurs LSP consécutifs ;
- $C_{s_{\max}}$, l'ensemble des chemins constitués d'au maximum $s_{\max}$ LSP consécutifs.

Voici les variables que nous utiliserons pour les différentes modélisations :

- x_{ab}, une variable entière binaire qui vaut 1 si et seulement si la nouvelle connexion utilise le LSP (a,b) et 0 sinon ;
- y_{ij}, une variable entière binaire qui vaut 1 si et seulement si la nouvelle connexion utilise le lien physique (i,j) et 0 sinon ;
- F_{ij}, le flot en bps du lien (i,j) ;
- $\overline{F_{ij}}$, le flot en bps du lien (i,j) si le nouveau trafic utilise le lien (i,j) ;
- $D_{ij} = d_{ij}(F_{ij})$, le délai en secondes du lien (i,j) qui est calculé en se basant sur un modèle de file d'attente M/M/1 ou M/M/1/k ;
- $\overline{D_{ij}} = d_{ij}(\overline{F_{ij}})$;
- $\Delta D_{ij} = \overline{D_{ij}} - D_{ij}$;
- D_{ab}, le délai en secondes du LSP (a,b) calculé en sommant les délai des liens qui constituent le LSP ;

- $\overline{D}_{ab}$, le délai en secondes du LSP (a,b) s'il est emprunté par la nouvelle connexion ;
- $\overline{D}_c$, le délai en secondes du chemin c (qui peut comporter jusqu'à $s_{\max}$ LSP) s'il est emprunté par la nouvelle connexion ;
- $D_c^{\max}$, le délai maximal que peut tolérer le chemin *c*. Il est égal au minimum des maxima de délais tolérés par les connexions empruntant exclusivement le chemin *c*. Si aucune connexion n'emprunte un chemin donné, $D_c^{\max} = \infty$;
- $P_{ij} = p_{ij}(F_{ij})$, la probabilité de perte de paquets du lien (i,j) qui est calculé en se basant sur un modèle de file d'attente M/M/1/k ;
- $\overline{P}_{ij} = p_{ij}(\overline{F_{ij}})$;
- $R_{ij} = r_{ij}(F_{ij}) = 1 - P_{ij}$, la probabilité que l'envoi d'un paquet soit réussi sur le lien (i,j) ;
- R_{ab}, la probabilité de réussite de l'envoi d'un paquet sur le LSP (a,b) calculée en multipliant les probabilité de réussite des liens qui constituent le LSP ;
- $\overline{R}_{ab}$, la probabilité de réussite de l'envoi d'un paquet sur le LSP (a,b) s'il est emprunté par la nouvelle connexion ;
- $\overline{P}_{ab} = 1 - \overline{R}_{ab}$, la probabilité de perte de paquets sur le LSP (a,b) s'il est emprunté par la nouvelle connexion ;
- R_c, la probabilité de réussite de l'envoi d'un paquet sur le chemin c (qui peut comporter jusqu'à $s_{\max}$ LSP) ;
- $\overline{R_c}$, la probabilité de réussite de l'envoi d'un paquet sur le chemin c (qui peut comporter jusqu'à $s_{\max}$ LSP) s'il est emprunté par la nouvelle connexion ;
- $P_c^{\max}$, la probabilité de perte maximale que peut tolérer le chemin *c*. Elle est égale au minimum des maxima des probabilités de perte tolérées par les connexions empruntant exclusivement le chemin *c*. Si aucune connexion n'emprunte un chemin donné, $P_c^{\max} = 1$;

- $R_c^{\min} = 1 - P_c^{\max}$, la probabilité de réussite minimale que peut tolérer le chemin *c*. Si aucune connexion n'emprunte un chemin donné, $R_c^{\min} = 0$.

Voici les constantes utilisées :

- y_{ij}^t, une constante 0-1 qui vaut 1 si et seulement si le trafic *t* utilise le lien (i, j) ;
- z_{ij}^{ab}, une constante 0-1 qui vaut 1 si et seulement si le LSP (a, b) utilise le lien (i, j) ;
- α^t, la quantité de flot de la connexion $t \in T$ en bps ($\alpha^t > 0$) ;
- α, la quantité de flot de la nouvelle connexion en bps ($\alpha > 0$) ;
- β^t, le délai maximal en secondes pouvant être supporté par le trafic $t \in T$ ($\beta^t > 0$) ;
- β, le délai maximal en secondes pouvant être supporté par la nouvelle connexion ($\beta > 0$) ;
- ϕ^t, la perte maximale pouvant être supportée par le trafic $t \in T$ ($1 > \phi^t > 0$) ;
- ϕ, la perte maximale pouvant être supportée par la nouvelle connexion ($1 > \phi > 0$) ;
- l_p, la longueur moyenne des paquets en bits ;
- l_{ij}, la longueur en Km du lien (i,j) ;
- v, le temps de propagation égal à 5 microsecondes par km ;
- C_{ij}, la capacité en bps du lien (i,j) ;
- $s_{\max}$, le nombre maximal de sauts qu'une connexion peut emprunter de l'origine à la destination ;
- g_{ij}, le coût fixe du lien (i,j) ;
- g_{ab}, le coût fixe du LSP (a,b) correspondant à la somme des coûts fixes g_{ij} des liens constituant le LSP.

4.1.3 Délai et perte de paquets

Pour le cas M/M/1, seul le délai est considéré et la formule de *Little* est utilisée :

$$D_{ij} = \frac{l_p}{C_{ij} - F_{ij}} + l_{ij} v, \quad \forall (i,j) \in M \tag{4.1}$$

avec $$F_{ij} < C_{ij} \quad \forall (i,j) \in M$$

Dans cette formule et dans le reste du travail, le délai considéré comprend le temps d'attente, le temps de transmission et le temps de propagation et nous négligeons le délai de traitement.

La Figure (4.1) présente la courbe du délai en fonction de l'utilisation des liens pour des files M/M/1 avec une taille de paquets de 1500 octets et une capacité de 100 Mbps. On constate que le délai est faible jusqu'à la zone de saturation de lien. A partir de là, le délai croit rapidement vers l'infini.

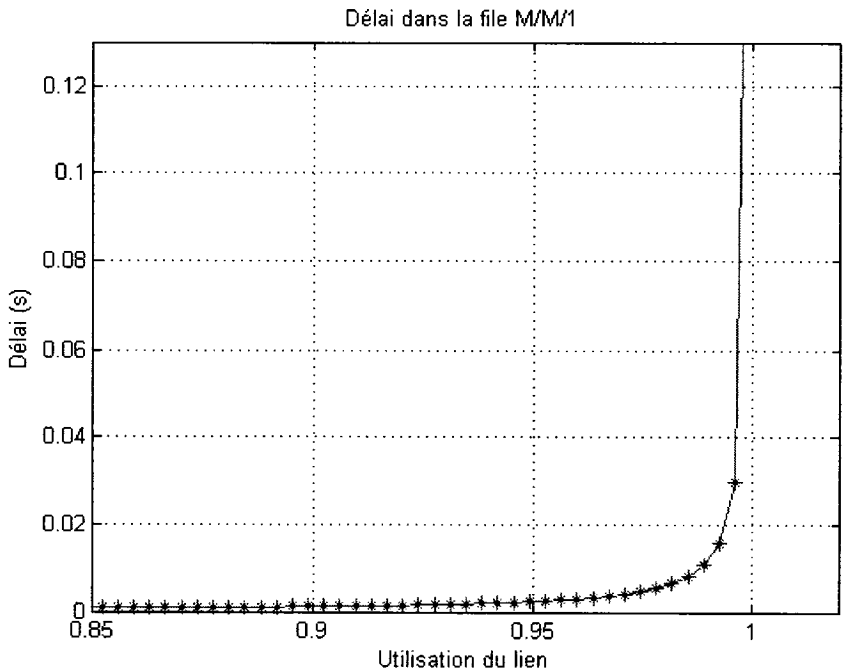

Figure 4.1 Délai dans une file M/M/1

Si nous utilisons un modèle M/M/1/k, la probabilité qu'un paquet trouve k éléments dans le système est donnée par :

$$P_k = \frac{\rho^k (1-\rho)}{1-\rho^{k+1}} \tag{4.2}$$

où pour chaque lien (i, j), $\rho = \frac{F_{ij}}{C_{ij}}$ est le taux d'utilisation du lien. Mentionnons que pour un système avec une taille de file valant *k-1*, P_k représente le taux de perte de paquets qui est une métrique que nous étudions. En effet, si le système contient *k-1* paquets dans la file et un paquet en traitement, le paquet qui arrive est rejeté.

Le délai sur un lien pour le modèle M/M/1/k est donné par la formule :

$$D_{ij} = \frac{\rho[1 + k\rho^{k+1} - (k+1)\rho^k]}{\lambda_a(1-\rho)(1-\rho^{k+1})}, \quad \forall (i,j) \in M \tag{4.3}$$

avec $\lambda_a = \lambda(1 - P_k)$

$\lambda = \frac{F_{ij}}{l_p}$ est le taux d'arrivée dans le système et λ_a représente la proportion du taux d'arrivée qui a été admise dans le système. Ce délai peut être reformulé pour donner :

$$D_{ij} = \frac{\rho[1 + k\rho^{k+1} - (k+1)\rho^k]}{\lambda(1-\rho)(1-\rho^k)}, \quad \forall (i,j) \in M \tag{4.4}$$

Les Figures (4.2) et (4.3) présentent respectivement les courbes du délai et de la perte de paquets pour une file M/M/1/k avec une taille de paquets de 1500 octets et une capacité de 100 Mbps sur chaque lien. On remarque que dans la zone de saturation du lien, la variation de délai dépend de la valeur de *k*. Plus *k* est grand, plus la variation de délai est importante. Le modèle M/M/1/k tolère un dépassement de capacité et on observe que le délai est borné. Cela résulte du fait que la taille de la file d'attente est limitée. Pour la perte de paquets, on remarque quand même que la perte de paquets commence à croître dans la zone de saturation. Cette perte est bornée supérieurement par 1 car c'est une probabilité.

Dans le modèle M/M/1/k, le flot théorique entrant sur le lien peut être supérieur à la capacité et cela se traduira par des pertes de paquets. A cause des contraintes sur la perte de paquets, le dépassement maximal de capacité sera très faible. Par contre, si dans une expérience la perte de paquets n'était pas considérée, il ne faudrait pas que le flot puisse dépasser la capacité car les résultats seraient biaisés. En effet, le délai du système

M/M/1/k étant borné, on accepterait du trafic même si les liens sont extrêmement saturés, ce qui n'est pas réaliste.

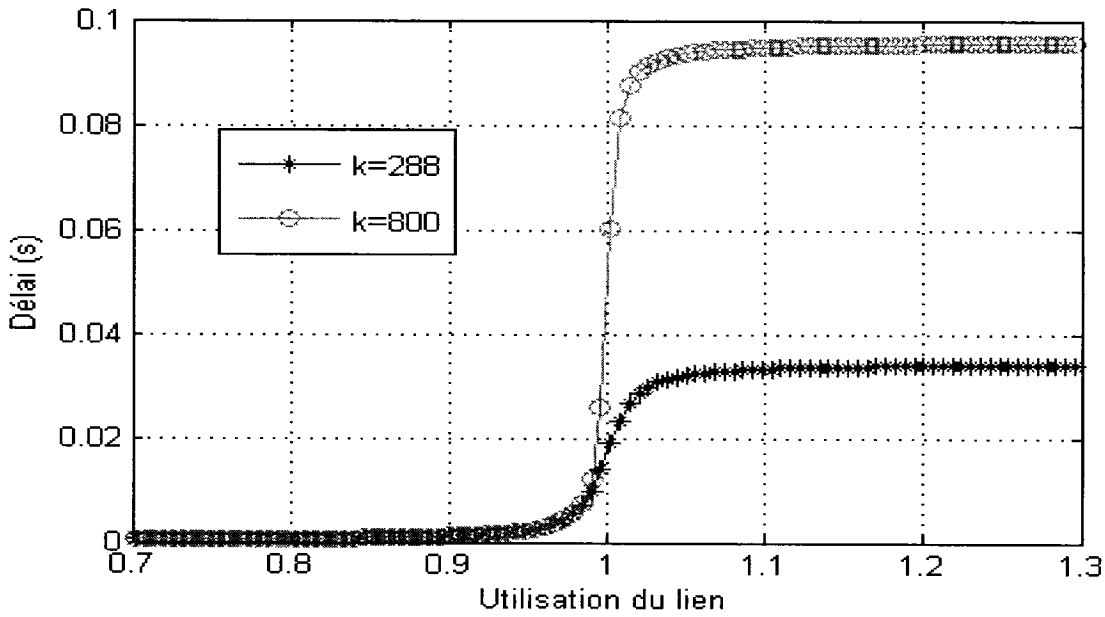

Figure 4.2 Délai dans une file M/M/1/k

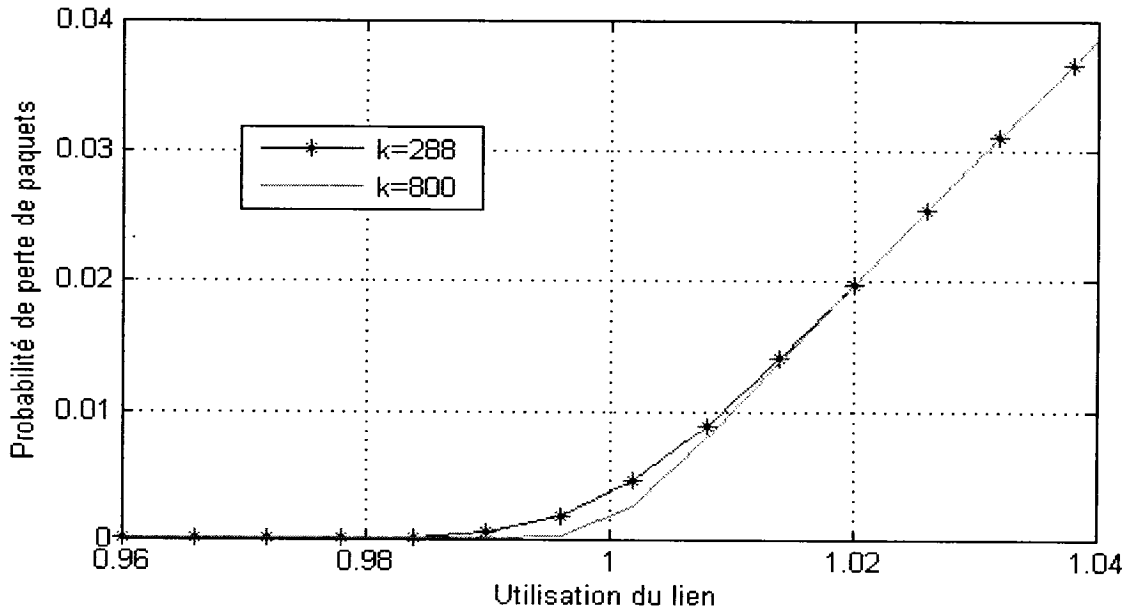

Figure 4.3 Probabilité de perte de paquets dans une file M/M/1/k

4.1.4 Modèles de base

Les modèles de base, notés $B_{x,y,k}$, ont la particularité qu'ils n'incluent que les contraintes de QdS concernant la connexion qui requiert l'accès au réseau. L'indice x donne une information sur le type d'objectif qui est recherché. Nous définissons $x = d$ pour l'objectif visant à minimiser le délai de bout en bout, $x = p$ lorsque l'objectif est d'utiliser le chemin avec la plus faible probabilité de perte de paquets et $x = f$ pour l'objectif visant à minimiser une somme de coûts fixe des liens utilisés par la connexion. Par ailleurs, $y = d$ indique que la contrainte de délai de bout en bout est considérée, $y = p$ indique que la contrainte de perte de paquets de bout en bout est prise en compte et $y = dp$ considère les deux types de contraintes. Enfin, l'indice k indique la longueur de la file d'attente considérée dans un modèle M/M/1/k. Lorsque cet indice est omis, il s'agit d'une file de longueur infinie basée sur un modèle M/M/1.

Il faut aussi mentionner que dans le cas d'objectifs avec des coefficients variables comme le délai, certains modèles essaient de trouver le plus court chemin en ne considérant que la charge du réseau avant le routage de la nouvelle connexion. Dans notre cas, nous recherchons le meilleur chemin en considérant la quantité de trafic ajoutée par la nouvelle connexion. Le modèle de base pour un cas M/M/1 ayant pour objectif de minimiser le délai est le suivant :

$B_{d,d,}$:

$$\min_{\{x_{ab}:(a,b)\in L\}} \sum_{(a,b)\in L} \overline{D_{ab}} x_{ab} \tag{4.5}$$

sujet à

$$\overline{F_{ij}} = (\sum_{t\in T} \alpha^t y_{ij}^t) + \alpha \sum_{(a,b)\in L} x_{ab} z_{ij}^{ab} \quad \forall (i,j) \in M \tag{4.6}$$

$$\overline{F_{ij}} < C_{ij} \quad \forall (i,j) \in M \tag{4.7}$$

$$\overline{D_{ij}} = \frac{l_p}{C_{ij} - \overline{F_{ij}}} + l_{ij} v, \quad \forall (i,j) \in M \tag{4.8}$$

$$\overline{D_{ab}} = \sum_{(i,j)\in M} \overline{D_{ij}} z_{ij}^{ab} \quad \forall (a,b) \in L \tag{4.9}$$

$$\sum_{(a,b)\in L} x_{ab} - \sum_{(a,b)\in L} x_{ba} \begin{cases} = 1 & \text{si b = destination} \\ = -1 & \text{si b = origine} \\ = 0 & \text{pour les autres cas} \end{cases} \quad (4.10)$$

$$x_{ab} \in \{0,1\} \quad \forall (a,b) \in L \quad (4.11)$$

$$\sum_{(a,b)\in L} \overline{D_{ab}} x_{ab} \leq \beta \quad (4.12)$$

L'expression (4.5) définit l'objectif qui consiste à minimiser le délai pour chaque nouvelle connexion en fonction des LSP utilisés. Quelque soit le modèle considéré, on peut remplacer la variable $\overline{D_{ab}}$ dans l'objectif par la constante g_{ab} pour avoir des coefficients fixes. Les coefficients g_{ab} sont calculés pour chaque LSP $(a,b) \in L$ en sommant les coûts fixes g_{ij} des liens composant le LSP. Ce sont des valeurs statiques. Les contraintes (4.6) permettent de calculer le flot sur chaque lien (i, j) qui est égal à la somme du flot de chaque connexion utilisant le lien additionnée du flot de la nouvelle connexion si celle-ci emprunte le lien (i, j). Le respect de la limite de capacité est garanti par les contraintes (4.7). Par ailleurs, les contraintes (4.8) et (4.9) permettent de calculer les délais projetés des liens et des LSP. En outre, les contraintes (4.10) permettent la conservation de flot pendant que les contraintes (4.11) assurent l'intégralité des variables x_{ab}. Enfin, la contrainte (4.12) garantie le respect de la contrainte de délai pour la connexion courante.

Les contraintes (4.8) introduisent une non linéarité dans le modèle $B_{d,d}$. Par ailleurs, ce type de problème a été démontré NP-difficile (Garey et Johnson, 1979). Ces mêmes observations sont valables quand on considère les files M/M/1/k et les pertes de paquets.

4.1.5 Linéarisation des modèles

Pour réduire la complexité du modèle de base, nous proposons une méthode de linéarisation basée sur une nouvelle formulation du problème. Tout d'abord, nous faisons l'hypothèse qu'une connexion n'emprunte qu'une seule fois un lien physique afin d'éviter les boucles dans le réseau. Le principe est de calculer le flot potentiel de chacun

des liens sur le réseau si ces derniers sont utilisés par la nouvelle connexion ($\overline{F_{ij}} = (\sum_{t \in T} \alpha^t y_{ij}^t) + \alpha \quad \forall (i,j) \in M$). Avec cette valeur, on peut calculer le délai et la perte de paquets potentiels des différents liens, ce qui permettra de trouver les mêmes paramètres pour les LSP. Ensuite, le modèle est bâti avec ces paramètres et il n'y a une mise à jour* des flots que si la nouvelle connexion utilise réellement un lien (i, j) donné. Les sections suivantes permettront de mieux comprendre la linéarisation. Les modèles proposés seront notés $P_{x,y,k}$, et suivront la même nomenclature que le modèle de base. À chaque itération, on suppose que pour chaque lien (i, j), le flot F_{ij}, le délai D_{ij} et le taux de réussite R_{ij} provenant de l'itération précédente sont connus.

4.1.6 Modèle pour file M/M/1 avec contraintes de délai

Dans le modèle $P_{d,d}$, lorsqu'une requête arrive au contrôleur, celui-ci commence par calculer les délais potentiels de chaque lien du réseau. À partir des valeurs obtenues, le modèle est généré puis résolu. Finalement, la mise à jour des constantes est effectuée en fonction des résultats de la résolution du modèle. Ces opérations sont détaillées dans les sections suivantes.

4.1.6.1 Calcul des délais

Pour chaque lien $(i,j) \in M$, calculer

$$\overline{F_{ij}} = (\sum_{t \in T} \alpha^t y_{ij}^t) + \alpha$$

Si $\overline{F_{ij}} < C_{ij}$, calculer

$$\overline{D_{ij}} = \frac{l_p}{C_{ij} - \overline{F_{ij}}} + l_{ij} v$$

Sinon

$$\overline{D_{ij}} = \infty$$

$$\Delta D_{ij} = \overline{D_{ij}} - D_{ij}$$

Pour chaque LSP $(a,b) \in L$, calculer

$$\overline{D_{ab}} = \sum_{(i,j)\in M} \overline{D_{ij}} z_{ij}^{ab}$$

Pour chaque chemin $c \in C_2$, calculer

$$\overline{D_c} = \sum_{(a,b)\in c} \overline{D_{ab}} \quad .$$

4.1.6.2 Génération du modèle

Modèle $P_{d,d}$

$P_{d,d}$:

$$\min_{\{x_{ab}:(a,b)\in L\}} \sum_{(a,b)\in L} \overline{D_{ab}} x_{ab} \tag{4.13}$$

sujet à

$$y_{ij} = \sum_{(a,b)\in L} z_{ij}^{ab} x_{ab} \quad \forall (i,j) \in M \tag{4.14}$$

$$0 \le y_{ij} \le 1 \quad \forall (i,j) \in M \tag{4.15}$$

$$\sum_{(a,b)\in L} x_{ab} - \sum_{(a,b)\in L} x_{ba} \begin{cases} = 1 & \text{si b = destination} \\ = -1 & \text{si b = origine} \\ = 0 & \text{pour les autres cas} \end{cases} \tag{4.16}$$

$$x_{ab} \in \{0,1\} \quad \forall (a,b) \in L \tag{4.17}$$

$$\sum_{(a,b)\in L} x_{ab} \le s_{\max} \tag{4.18}$$

$$\sum_{(a,b)\in L} \overline{D_{ab}} x_{ab} \le \beta \tag{4.19}$$

$$\sum_{(i,j)\in M} (D_{ij} + y_{ij} \Delta D_{ij}) y_{ij}^t \le \beta^t \quad \forall t \in T \tag{4.20}$$

L'objectif (4.13) reste le même que celui du modèle de base. Les contraintes (4.14) et (4.15) assurent que la nouvelle connexion ne passe qu'une seule fois par un lien afin de ne pas avoir de boucles sur le chemin. Notons qu'à cause des contraintes (4.14) et (4.17),

les variables y_{ij} sont entières. D'autre part, les contraintes (4.16) et (4.17) jouent respectivement le même rôle que les contraintes (4.10) et (4.11).

Pour réduire la complexité du problème, nous avons introduit les contraintes (4.18) qui permettent de limiter le nombre de LSP qu'une connexion peut emprunter de l'origine à la destination. Cette limitation est réaliste car un LSP peut être composé de plusieurs liens. Nous démontrerons que cette limitation n'a pas d'effet significatif sur le blocage. La contrainte (4.19) permet de respecter une limite en termes de délai maximal pour la connexion qui demande l'accès.

Par ailleurs, les contraintes (4.20) permettent de respecter le délai maximal pour chaque connexion qui est déjà en service sur le réseau. ΔD_{ij} n'est ajouté que si la nouvelle connexion utilise le lien (i, j). Aussi, dans ce modèle, les contraintes de capacités ne sont pas mentionnées. Toutefois, chaque lien où il y a un dépassement de capacité à un délai infini ce qui entraîne qu'il ne sera jamais sélectionné dans la solution à cause de la contrainte (4.19).

Réduction du nombre de contraintes

A ce stade, les contraintes (4.20) peuvent être nombreuses car leur nombre équivaut au nombre de connexions en service sur le réseau. Pour simplifier cela, nous avons fixé $s_{\max} = 2$. Nous prouverons que le blocage ne varie pas significativement lorsque $s_{\max}$ prend les valeurs 2, 3 ou 8. Avec $s_{\max} = 2$, nous pouvons utiliser une approche par chemin car chaque chemin comprendra au maximum 2 LSP. Ce procédé nous permet de réduire le nombre de contraintes car il est possible que plusieurs connexions utilisent le même chemin.

Nous créons un tableau qui conserve le minimum des maxima de délais autorisés pour les connexions empruntant un seul LSP ou deux LSP consécutifs ($D_c^{\max}$). En bâtissant notre modèle, on calcule pour chaque LSP et pour chaque paire de LSP consécutifs le délai potentiel $\overline{D_c} = \sum_{(a,b)\in c} \overline{D_{ab}} \quad \forall c \in C_2$. Si cette somme est supérieure au

maximum autorisé ($\overline{D_c} \geq D_c^{\max}$), la contrainte est écrite pour le chemin car il y a un risque potentiel de dépassement. Dans le cas contraire, il ne sert à rien d'écrire une contrainte pour ce chemin car, dans le pire cas, le délai maximal ne sera pas dépassé. Cela permet de réduire le nombre de contraintes à écrire. Les contraintes (4.20) pourront être remplacées par les contraintes (4.21) ci-dessous:

$$\sum_{(a,b)\in c}\sum_{(i,j)\in M}(D_{ij} + y_{ij}\Delta D_{ij})z_{ij}^{ab} \leq D_c^{\max}$$
$$\forall c \in C_2, \overline{D_c} \geq D_c^{\max} \qquad (4.21)$$

Les contraintes (4.21) permettent donc de respecter les contraintes de délais pour chaque chemin de un ou deux LSP. ΔD_{ij} n'est ajouté que si la nouvelle connexion utilise le lien (i, j).

Le modèle proposé est NP-difficile. Cependant, nous montrerons que la linéarité et le faible nombre de variables permet de le résoudre rapidement.

4.1.6.3 Mise à jour

Si la résolution du modèle fournit une solution réalisable, les informations permettant d'indexer le chemin de la nouvelle connexion sont conservées et la mise à jour des délais maxima autorisés pour les chemins est faite. On désigne la nouvelle connexion par $t' \in T$. Voici les opérations à effectuer seulement si la connexion est acceptée :

Pour chaque lien $(i, j) \in M$,

Si la nouvelle connexion utilise le lien $(i, j) \in M$, alors

$y_{ij}^{t'} = 1$

Sinon

$y_{ij}^{t'} = 0$

Pour le chemin $c \in C$ utilisé par $t' \in T$,

Si $\beta^{t'} < D_c^{\max}$, alors

$D_c^{\max} = \beta^{t'}$

4.1.7 Modèle pour file M/M/1/k avec contraintes de délai

Le modèle $P_{d,d,k}$ a beaucoup de similarités avec le modèle $P_{d,d}$ et les étapes de résolutions sont les mêmes que celles de la section précédente.

4.1.7.1 Calcul des délais

Pour chaque lien $(i,j) \in M$, calculer

$$\overline{F_{ij}} = (\sum_{t \in T} \alpha^t y_{ij}^t) + \alpha$$

Si $\overline{F_{ij}} < C_{ij}$, calculer

$$\lambda = \frac{\overline{F_{ij}}}{l_p}, \quad \rho = \frac{\overline{F_{ij}}}{C_{ij}}, \quad \overline{D_{ij}} = \frac{\rho[1 + k\rho^{k+1} - (k+1)\rho^k]}{\lambda(1-\rho)(1-\rho^k)}$$

Sinon

$$\overline{D_{ij}} = \infty$$

$$\Delta D_{ij} = \overline{D_{ij}} - D_{ij}$$

Pour chaque LSP $(a,b) \in L$, calculer

$$\overline{D_{ab}} = \sum_{(i,j) \in M} \overline{D_{ij}} z_{ij}^{ab}$$

Pour chaque chemin $c \in C_2$, calculer

$$\overline{D_c} = \sum_{(a,b) \in c} \overline{D_{ab}} \quad .$$

4.1.7.2 Génération du modèle

Modèle $P_{d,d,k}$

Le modèle généré est le même que celui de $P_{d,d}$. à la section 4.1.6.2.

Réduction du nombre de contraintes

Les mêmes processus décrits dans la section 4.1.6.2 sont utilisés pour réduire le nombre de contraintes.

4.1.7.3 Mise à jour

Nous appliquons les mêmes opérations de mise à jour que celles décrites dans la section 4.1.6.3.

4.2 Évaluation de performance

Nous avons évalué nos modèles sur des machines SUN Linux disposant de 8 GB de RAM avec un CPU de 2.4 GHz. Nous considérons qu'un contrôleur dédié au contrôle d'admission dans un réseau pourrait avoir des capacités égales ou supérieures. Le programme est écrit en C et les modèles sont résolus en utilisant le logiciel CPLEX 9.1.3.

4.2.1 Jeux de données pour les tests

Nous avons utilisé des réseaux 2-connexes générés aléatoirement. La taille des réseaux varie de 5 à 80 nœuds. Les réseaux sont créés en construisant d'abord un cycle hamiltonien puis des liens sont ajoutés entre les paires de nœuds jusqu'à ce qu'on atteigne le nombre de liens voulu. Les paires de nœuds choisies pour les liens sont équiprobables. Par ailleurs, les paires de nœuds pour le trafic sont choisies aléatoirement et pour chaque connexion, chacun des nœuds a la même probabilité d'être origine ou destination. Comme il est important pour nous d'étudier les réseaux dans des contextes de charge élevée, le nombre total de connexions pour chaque réseau est fixé à $80n$, donc 80 fois le nombre de nœuds. La quantité de trafic pour chaque flot est de 1 Mbps. D'autre part, les délais et pertes de paquets maxima autorisés pour les connexions sont respectivement pris de façon aléatoire dans les ensembles {50, 100, 150, 200, 250, 300} ms et {1, 2, 3, 4, 5} %. Lors des simulations, les connexions sont traitées séquentiellement comme le ferait un vrai contrôleur. C'est pour cela qu'il est important que la durée d'exécution totale de nos algorithmes soit relativement petite.

Par ailleurs, pour les files M/M/1/k, en nous basant sur les spécifications des équipements CATALYST 6500 de CISCO, nous avons utilisé les tailles de mémoire de 432 et 1200 Ko, ce qui équivaut à $k = 288$ et $k = 800$ paquets. Le tableau 4.1 présente les réseaux de tests.

Tableau 4.1 Jeux de données pour les tests

Nombre de nœuds	Nombre de nœuds frontières	Nombre de liens	Nombre de connexions
5	3	10	400
10	6	15	800
20	12	30	1600
30	18	45	2400
40	24	60	3200
50	30	75	4000
60	36	90	4800
70	42	105	5600

4.2.2 Mesures de performances

Les mesures de performance qui nous intéressent sont :

- le taux de blocage pour savoir comment l'ajout des nouvelles contraintes influence l'acceptation des connexions sur le réseau ;
- l'impact sur le blocage de la limitation du nombre maximal de LSP pouvant être utilisé par une connexion ;
- les valeurs moyennes de délai et de perte de paquets pour vérifier que les nouvelles contraintes permettent de réduire ces valeurs ;
- la proportion de connexions en service ayant excédé les limites de QdS fixées ;
- le temps d'exécution pour résoudre les modèles avec le logiciel CPLEX afin de juger de la complexité des modèles proposés ;
- le temps total d'exécution qui regroupe le prétraitement, la résolution du modèle et la mise à jour des données.

Pour nos simulations, nous allons comparer les résultats obtenus avec les modèles proposés *P* à ceux obtenus avec les modèles *R* (référence). Les modèles *R* correspondent au cas classique de CAC où l'objectif, pour une novelle onnexion, est de minimiser un

paramètre donné (délai, perte de paquets ou coûts fixes) en ne garantissant les paramètres de délai et/ou de perte de paquets de bout en bout qu'à la nouvelle connexion. Pour mieux évaluer l'impact des nouvelles contraintes introduites, nous allons aussi imposer aux modèles de références, une limitation de 2 LSP maximum utilisés par chemin. Nous verrons que cette limitation n'influe pas sur le blocage. Dans ce chapître, l'objectif utilisé est la minimisation du délai.

4.2.3 Impact du nombre maximal de LSP autorisés

Puisque nous limitons le nombre maximal de LSP autorisés à 2, nous devons juger de l'impact de cette limitation sur le blocage. Pour ce faire, nous allons évaluer les modèles en autorisant 2, 3 ou 8 LSP. Dans notre approche, la complexité va dépendre du nombre de LSP autorisés pour une connexion. L'évaluation de nos propositions avec 3 ou 8 LSP autorisés s'avère donc fastidieuse. Toutefois, puisque nous voulons comparer nos modèles *P* avec les modèles de références *R* utilisant 2 LSP, nous allons donc évaluer l'impact de la limitation à 2 LSP sur les modèles de références afin de vérifier la qualité des modèles de comparaison choisis.

4.2.3.1 File M/M/1

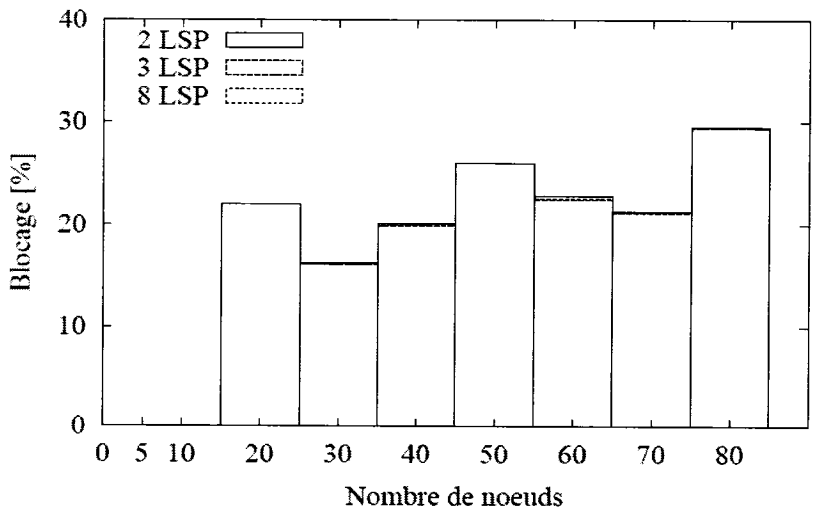

Figure 4.4 Taux de blocage pour une file M/M/1 en fonction de S_{max}

La Figure (4.4) présente le taux de blocage pour les connexions avec l'utilisation de files M/M/1. Nous observons que les résultats sont similaires pour 2, 3 et 8 LSP. Nous pouvons donc dire que la restriction sur le nombre de LSP ne dégrade pas le blocage.

4.2.3.2 File M/M/1/k

Les Figures (4.5a) et (4.5b) présentent le taux de blocage en fonction de S_{max} pour les connexions avec l'utilisation de files M/M/1/k pour k = 288 et k = 800. Les résultats montrent que le taux de blocage est similaire.

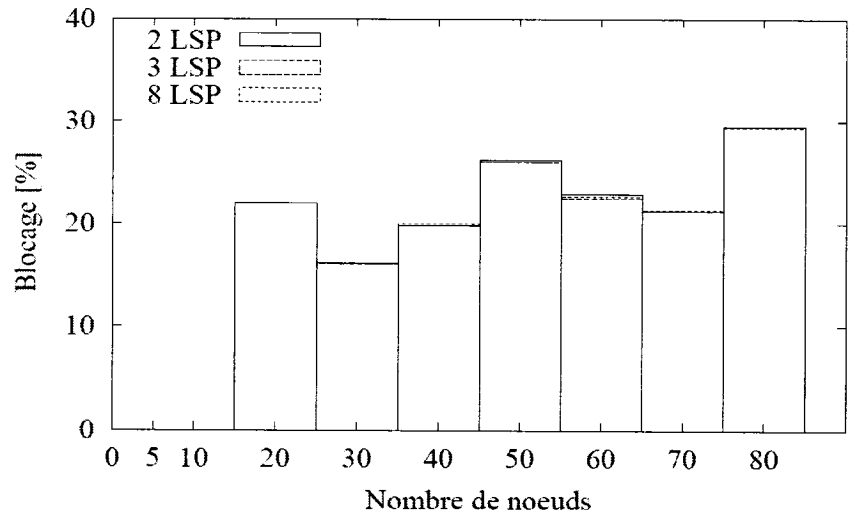

Figure 4.5a Blocage (k = 288)

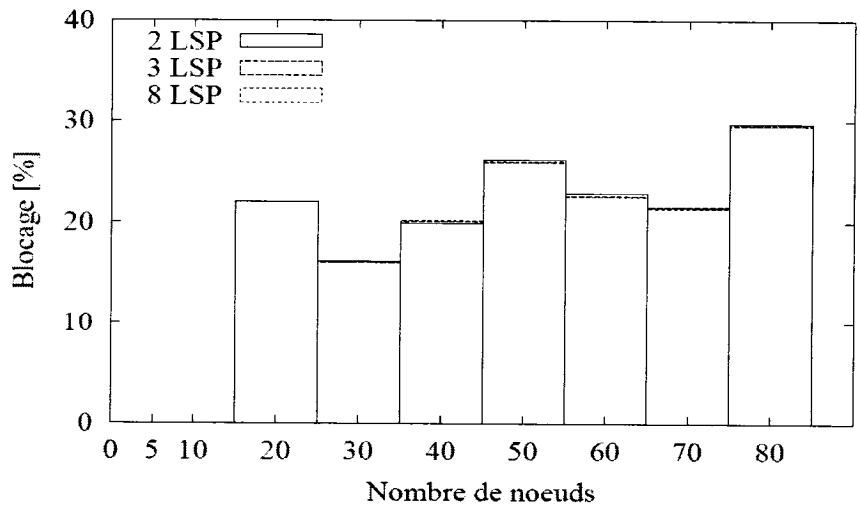

Figure 4.5b Blocage (k = 800)

4.2.3.3 Conclusions sur l'impact de S_{max}

La limite du nombre de LSP que nous avons imposée n'influe pas significativement sur le taux de blocage. On pourrait l'expliquer par le fait qu'un LSP comporte plusieurs liens et aussi que l'objectif est de minimiser le délai, ce qui est relié à la longueur du chemin. Nous pouvons donc dire que nos modèles de référence sont de bonnes bases de comparaison.

4.2.4 Blocage des connexions

Puisque nous ajoutons de nouvelles contraintes, il est important pour nous d'évaluer le taux de blocage des nouvelles connexions par rapport aux modèles de références.

4.2.4.1 File M/M/1

La Figure (4.6) présente le taux de blocage pour les connexions avec l'utilisation de files M/M/1. Nous observons que les résultats sont très proches. Nos contraintes n'ont donc pas d'effets négatifs sur le blocage car les différences sont en dessous de 2%.

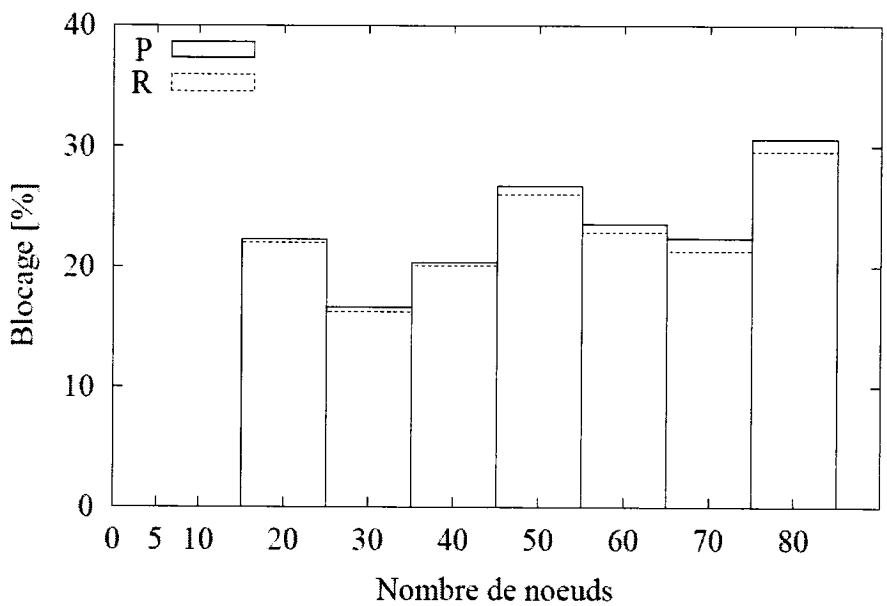

Figure 4.6 Taux de blocage pour une file M/M/1

4.2.4.2 File M/M/1/k

Les Figures (4.7a) et (4.7b) présentent le taux de blocage pour les connexions avec l'utilisation de files M/M/1/k pour k = 288 et k = 800. Pour k = 288, les différences sont en dessous de 2%. Lorsque k vaut 800 paquets, l'écart entre P et R ne dépasse jamais la valeur de 6%.

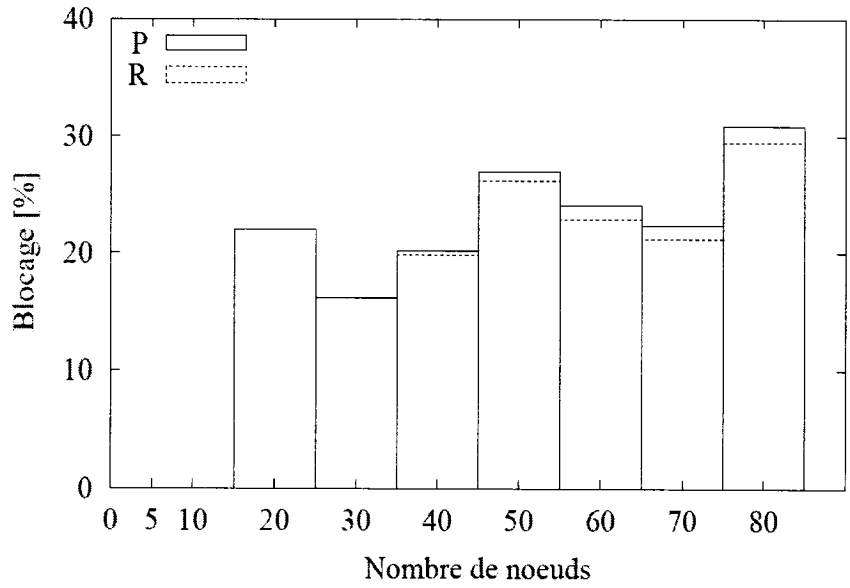

Figure 4.7a Blocage (k = 288)

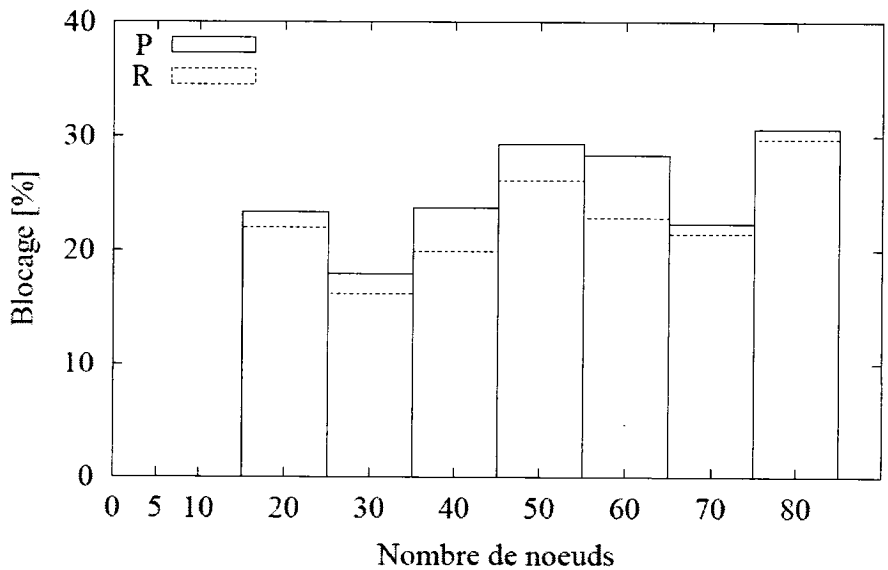

Figure 4.7b Blocage (k = 800)

4.2.4.3 Explications sur le blocage

Pour $k = 288$, les différences sont en dessous de 2%. Comme le délai dans les files est borné supérieurement et que la taille de la file est relativement petite, le fait d'être dans une zone de saturation du lien n'a pas un grand impact sur les connexions. En effet, la Figure (4.11a) nous montre que R procure des ratios de dépassement inférieur à 5.5% lorsque $k = 288$. C'est pourquoi la différence de blocage est faible.

Lorsque k vaut 800, l'écart entre P et R est inférieur à 6%. La différence s'explique par la taille des files d'attente et la borne supérieure sur le délai dans une file. Lorsque k est très petit, pratiquement aucune connexion n'excédera sa limite de délai dans la zone de saturation du lien et les contraintes que nous ajoutons dans P n'ont pas d'impact d'où un taux de blocage similaire. À l'extrême, si k est très grand, le délai croit rapidement dans la zone de saturation et toutes les connexions excèderont leur limite de délai dans cette zone. Le taux de blocage est presqu'identique car les deux modèles vont chercher à éviter cette zone de saturation car aucune connexion ne pourra être acceptée si elle passe par un lien qui est proche de la saturation. Nos contraintes auront donc un impact limité. Cette situation se retrouve dans le cas M/M/1 (Figure 4.6).

Dans les cas intermédiaires comme lorsque $k = 800$, la variation de délai dans la zone de saturation n'implique pas que toutes les connexions empruntant le lien excédent leur limite. En fait, dans notre cas, on peut supposer que les connexions ayant une limite de 50 ms auront plus de chances d'être en dépassement que les connexions ayant une limite de 300 ms. Les contraintes que nous rajoutons entrainent donc un peu plus de blocage. Toutefois, un maximum de 6% est une différence acceptable pour garantir les critères de QdS à chaque connexion établie.

4.2.5 Délai moyen de bout en bout

Notre modèle devrait réduire le délai moyen de bout en bout des connexions puisque la contrainte de bout en bout de chaque connexion est prise en compte. Nous allons vérifier cela en relevant pour chaque réseau la valeur du délai moyen.

4.2.5.1 File M/M/1

La Figure (4.8) présente le délai moyen encouru par les connexions lorsqu'un modèle M/M/1 est considéré. On voit que le modèle *P* permet de garder le délai moyen en dessous de 25 ms tandis que le modèle de référence *R* donne des délais dépassant les 400 ms. C'est un résultat qui n'est pas surprenant puisqu'avec nos propositions *P*, nous garantissons le délai de bout en bout de chaque connexion en service.

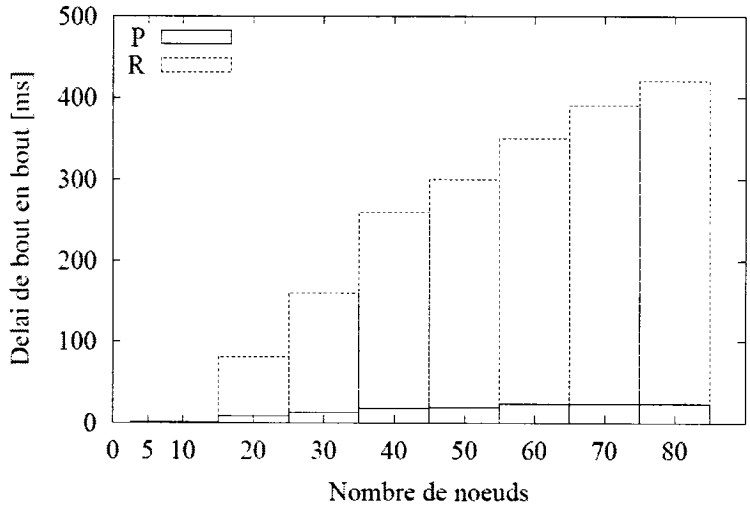

Figure 4.8 Délai de bout en bout pour une file M/M/1

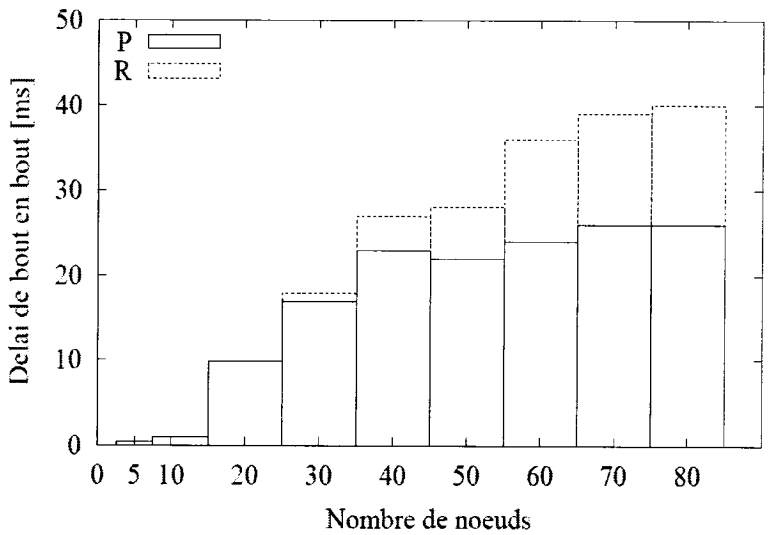

Figure 4.9a Délai (k = 288)

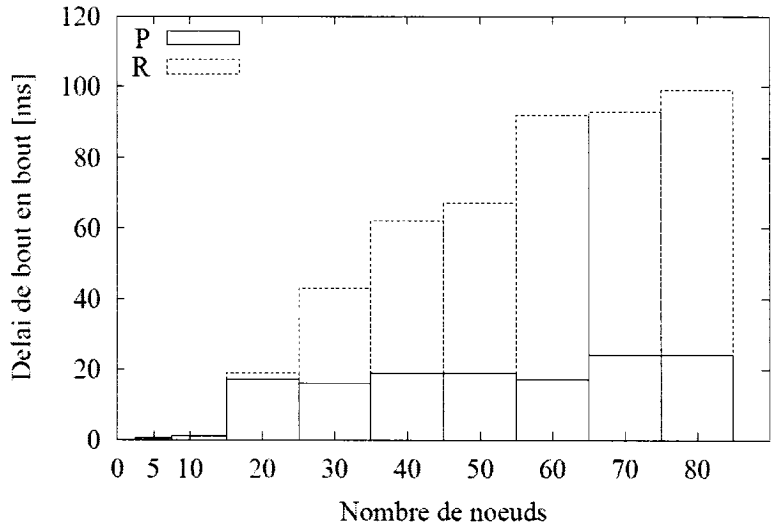

Figure 4.9b Délai (k = 800)

4.2.5.2 File M/M/1/k

En se référant aux Figures (4.9a) et (4.9b), on voit que pour $k = 288$, P maintient le délai moyen en dessous de 30 ms tandis que R atteint les 40 ms. Pour $k = 800$, le délai maximum obtenu avec P est de 22 ms comparativement à 100 ms pour R. Comme nous l'espérions, les résultats de P sont meilleurs.

4.2.5.3 Conclusion sur le délai moyen

Le modèle P permet d'obtenir des délais moyens nettement inférieurs à R. Les résultats du modèle R varient considérablement en fonction de la taille de la file. En fait, le modèle M/M/1 correspond à une taille de file infinie, ce qui entraîne des délais élevés. Si le réseau est fortement chargé, les files d'attentes sont pleines et plus elles sont grandes, plus il y aura de délai d'attente dans la file. Notre modèle évite cela en considérant les contraintes de délais pour chaque connexion.

4.2.6 Ratio de connexions ayant dépassées leur délai maximal

Nous allons maintenant évaluer le nombre de connexions qui dépassent la limite requise lorsque le modèle R est utilisé. Cela nous permet d'évaluer l'apport de P. Pour faire ce calcul de ratio, nous divisons le nombre de connexions ayant dépassées leur délai

par le nombre total de connexions admises sur le réseau.

4.2.6.1 File M/M/1

La Figure (4.10) présente le taux de connexions ayant excédé leur limite maximale de délai.

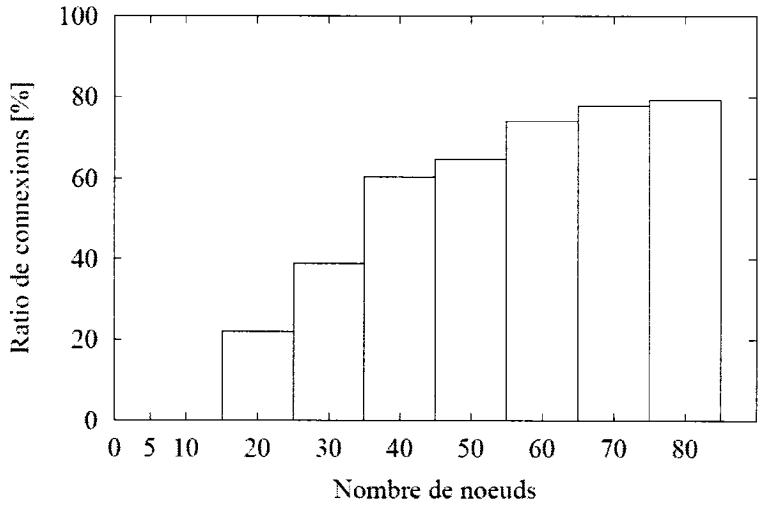

Figure 4.10 Ratio de connexions ayant dépassé leur délai maximal (M/M/1)

On voit que le modèle *R* peut causer jusqu'à 80% de dépassement. Cette valeur est aussi due au fait que la file M/M/1 modélise une file de taille infinie ce qui peut résulter en de grands délais.

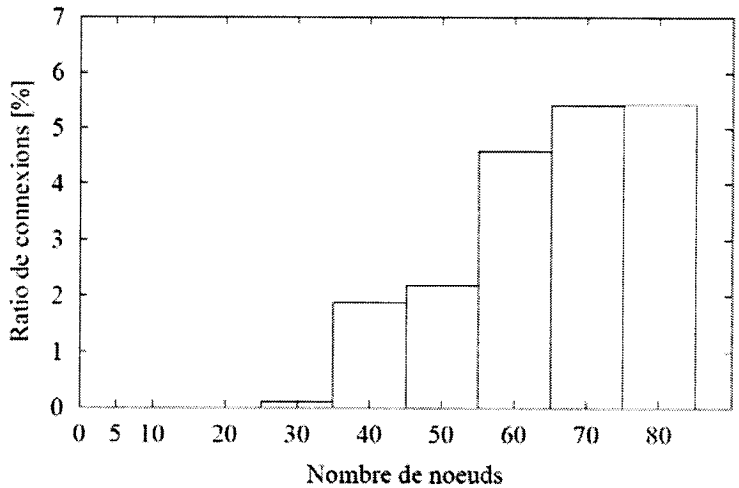

Figure 4.11a Ratio de dépassement du délai (k = 288)

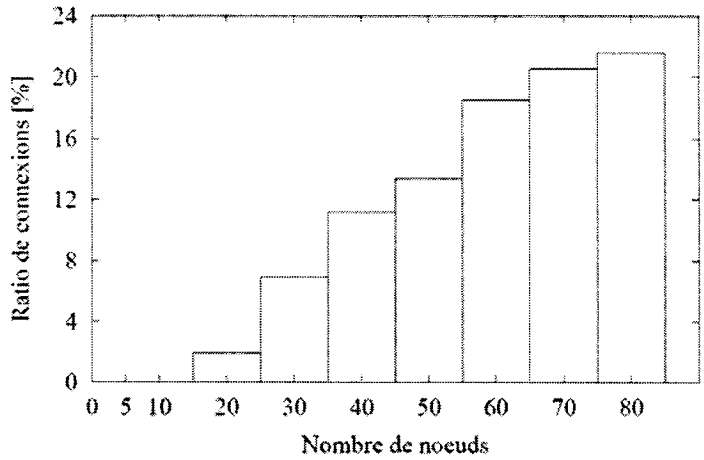

Figure 4.11b Ratio de dépassement du délai (k = 800)

4.2.6.2 File M/M/1/k

Les Figures (4.11a) et (4.11b) présentent le taux de connexions en situation de dépassement de leur limite maximale de délai pour des files M/M/1/k.

En se référant aux Figures (4.11a) et (4.11b), on voit que pour k = 288, il y a un maximum de 5.5% de connexions qui ne respectent pas leurs paramètres tandis que pour k = 800, il y a un maximum de 22% de connexions en situation de dépassement. Le modèle que nous proposons s'avère donc intéressant puisqu' il n'y a aucune connexion en situation de dépassement.

4.2.6.3 Conclusion sur le ratio de connexion dépassant leur limite de délai

P permet d'assurer qu'aucune connexion ne sera servie en deça des paramètres négociés au départ. On voit que les ratios obtenus avec R varient considérablement en fonction de la taille de la file. Cet état de fait est directement relié au délai moyen et au temps d'attente dans la file en situation de forte charge du réseau. En effet, le délai encouru dans une file lorsque k = 800 est supérieur au délai lorsque k = 288.

4.2.7 Temps d'exécution

Dans cette section, nous étudions le temps d'exécution CPU pour la résolution du modèle avec CPLEX et le temps total pour faire le CAC dynamique d'une connexion.

4.2.7.1 File M/M/1

La Figure (4.12) présente les temps d'exécution dans le cas d'un modèle M/M/1.

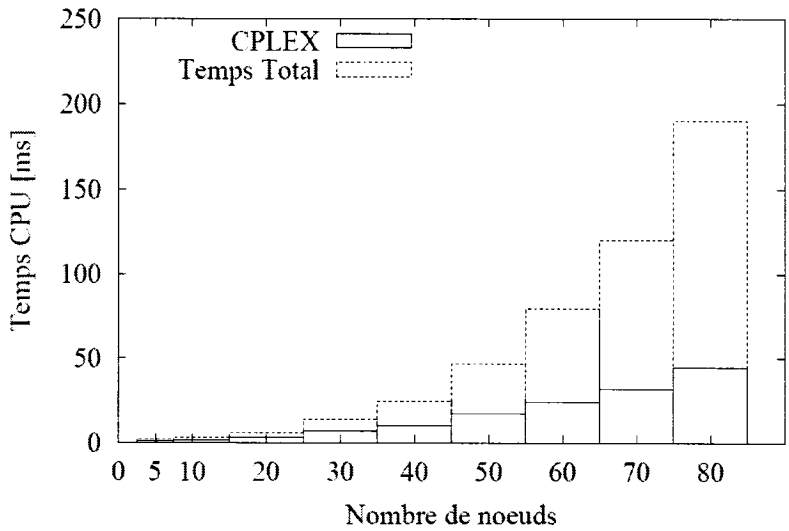

Figure 4.12 Temps d'exécution (M/M/1)

Les temps de résolution du modèle avec CPLEX sont en dessous de 50 ms tandis que le temps total pour faire le CAC est en dessous de 200 ms. Cela reste acceptable dans un cadre de contrôle d'admission dynamique de connexions ayant des requis de QdS.

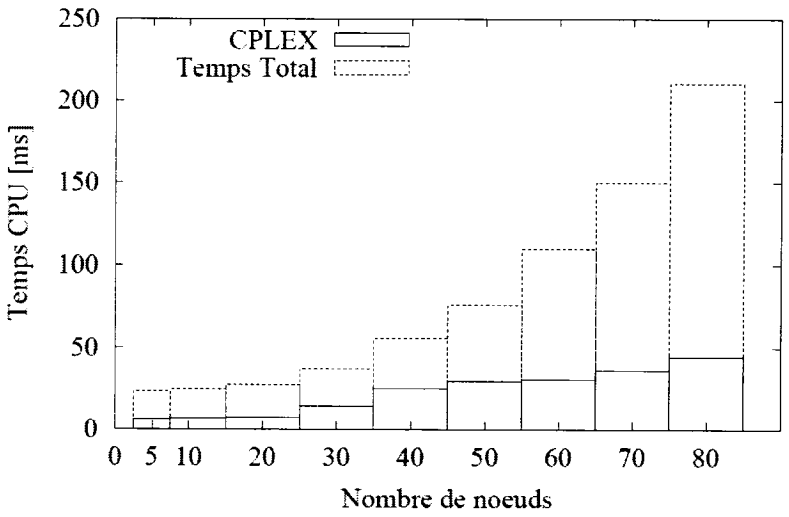

Figure 4.13a Temps d'exécution M/M/1/k (k = 288)

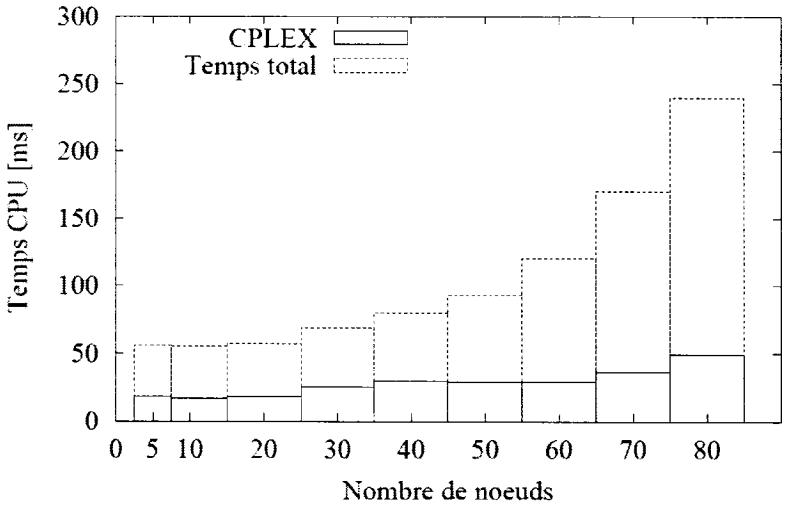

Figure 4.13b Temps d'exécution M/M/1/k (k = 800)

4.2.7.2 File M/M/1/k

La Figure (4.13a) montre que le temps de résolution du modèle avec CPLEX est en dessous de 50 ms pour k = 288 et k = 800. Au niveau du temps total d'exécution de l'algorithme pour de grands réseaux, nous constatons sur la Figure (4.13b) que nous ne dépassons pas 250 ms.

4.2.7.3 Conclusion sur le temps d'exécution

Bien que nous rajoutions des contraintes supplémentaires, nous sommes capables de trouver une solution à notre problème en moins d'un quart de seconde pour des problèmes de grandes tailles. La différence dans le temps total obtenu pour le cas M/M/1 et le cas M/M/1/k est due au fait que les calculs de délais sont plus complexes dans le cas M/M/1/k. Nous pouvons donc conclure que notre solution peut être utilisée dans des réseaux de grandes tailles et fournir des temps d'exécutions raisonnables.

4.2.8 Conclusions

Cette évaluation de performance prouve la qualité des algorithmes et modèles proposés. Nous avons montré que la limite du nombre de LSP imposée ne détériore pas le taux de blocage. La différence en termes de blocage par rapport aux modèles de

références est de moins de 6%. En considérant que le fait de ne pas bloquer ces 6 % de connexions conduit à dégrader le service de 22 % de connexions, cette augmentation du blocage est acceptable. D'autre part, les délais moyens que nous obtenons sont largement en dessous de ceux fournis par les modèles de base quel que soit le modèle de file d'attente considéré. Nous constatons donc que le délai de bout en bout est réduit tandis que les requis de chacune des connexions en service sont respectés. Enfin, le temps de résolution globale est acceptable puisque nous sommes en dessous de 250 ms pour chacun des scénarios. Ce temps d'exécution pourrait être réduit en optimisant le code et en utilisant des machines plus puissantes, ce qui serait le cas pour un contrôleur réel. Après avoir obtenu ces bons résultats pour la contrainte de délai, nous allons maintenant aborder la perte de paquets et les modèles multi-contraintes.

CHAPITRE V

PROBLÈME MULTI-CONTRAINTES DE ROUTAGE ET CONTRÔLE D'ADMISSION DES CONNEXIONS

Au chapitre IV, les résultats que nous avons obtenus nous ont permis de respecter les contraintes de délai pour toutes les connexions en service sur le réseau. Toutefois, dans le cadre d'applications avec des exigences de QdS, il est aussi important de garantir un taux maximal de perte de paquets. En effet, une application qui a un délai de transfert sur le réseau de 50 ms mais un taux de perte de paquets de 75% trouverait son fonctionnement considérablement altéré.

Sur un chemin donné, la perte de paquets est une mesure multiplicative. Il nous faut donc utiliser la fonction logarithmique pour rendre cette contrainte additive, ce qui facilite la résolution du problème. Par ailleurs, nous calculons la probabilité de succès de chaque lien et la probabilité de succès d'un LSP est le produit des probabilités de succès des liens composant le LSP. Ce même principe est appliqué pour la probabilité de succès d'un chemin composé de plusieurs LSP. En outre, la somme de la probabilité de succès et de la probabilité de perte sur un chemin doit donner 1.

Rappelons que les sections 4.1.1 à 4.1.5, notamment la définition des constantes et des variables, s'appliquent aussi pour ce présent chapitre. Pour pouvoir tenir compte de la perte de paquets, le modèle de file d'attente choisi est M/M/1/k car le modèle M/M/1 considère une file de taille infinie.

Dans ce chapitre, après avoir défini un modèle de base pour le problème multi-contraintes, nous exposons notre modèle mathématique pour le problème avec des contraintes de perte de paquets de bouts en bouts en explicitant les différentes contraintes. Suite à cela, nous présenterons notre proposition pour le problème multi-contraintes. Nous terminerons ce chapitre par une évaluation de performance des différents modèles proposés.

5.1 Modèle de base

Le modèle de base pour un cas M/M/1/k avec contraintes de délais et de perte de paquets et ayant pour objectif de minimiser le délai est le suivant :

$B_{d,dp,k}$:

$$\min_{\{x_{ab}:(a,b)\in L\}} \sum_{(a,b)\in L} \overline{D_{ab}} x_{ab} \tag{5.1}$$

Sujet à

$$\overline{F_{ij}} = (\sum_{t\in T} \alpha^t y_{ij}^t) + \alpha \sum_{(a,b)\in L} x_{ab} z_{ij}^{ab} \quad \forall (i,j) \in M \tag{5.2}$$

$$\overline{F_{ij}} < C_{ij} \quad \forall (i,j) \in M \tag{5.3}$$

$$\lambda = \frac{\overline{F_{ij}}}{l_p} \tag{5.4}$$

$$\rho = \frac{\overline{F_{ij}}}{C_{ij}} \tag{5.5}$$

$$\overline{D_{ij}} = \frac{\rho[1+k\rho^{k+1} - (k+1)\rho^k]}{\lambda(1-\rho)(1-\rho^k)} + l_{ij}v, \quad \forall (i,j) \in M \tag{5.6}$$

$$\overline{D_{ab}} = \sum_{(i,j)\in M} \overline{D_{ij}} z_{ij}^{ab} \quad \forall (a,b) \in L \tag{5.7}$$

$$\overline{P_{ij}} = \frac{\rho^k(1-\rho)}{1-\rho^{k+1}}, \quad \forall (i,j) \in M \tag{5.8}$$

$$\overline{R_{ij}} = 1 - \overline{P_{ij}}, \quad \forall (i,j) \in M \tag{5.9}$$

$$\overline{P_{ab}} = 1 - \prod_{(i,j)\in M : z_{ij}^{ab}=1} \overline{R_{ij}}, \quad \forall (a,b) \in L \tag{5.10}$$

$$\sum_{(a,b)\in L} x_{ab} - \sum_{(a,b)\in L} x_{ba} \begin{cases} = 1 & \text{si b = destination} \\ = -1 & \text{si b = origine} \\ = 0 & \text{pour les autres cas} \end{cases} \tag{5.11}$$

$$x_{ab} \in \{0,1\} \quad \forall (a,b) \in L \tag{5.12}$$

$$\sum_{(a,b)\in L} \overline{D_{ab}} x_{ab} \le \beta \tag{5.13}$$

$$\prod_{(a,b)\in L}(1-\overline{P_{ab}}x_{ab}) \geq (1-\phi) \tag{5.14}$$

Les expressions (5.1) à (5.3) sont identiques aux expressions (4.5) à (4.7). Les expressions (5.4) et (5.5) permettent de calculer respectivement le taux d'arrivée et le taux d'utilisation du lien. Par ailleurs, les contraintes (5.6) et (5.7) permettent de calculer les délais projetés des liens et des LSP. Les contraintes (5.8) et (5.9) calculent la probabilité de perte de paquets et la probabilité de succès sur un lien tandis que les contraintes (5.10) calculent la probabilité de perte sur un LSP.

En outre, les contraintes (5.11) permettent la conservation de flot pendant que les contraintes (5.12) assurent l'intégralité des variables x_{ab}. Enfin, les blocs de contraintes (5.13) et (5.14) garantissent le respect de la contrainte de délai et de la contrainte de perte de paquets pour la connexion courante. Pour les contraintes (5.14), mentionnons que la solution où $x_{ab} = 0 \quad \forall (a,b) \in L$ n'est pas réalisable à cause des contraintes (5.18) de conservation de flot. On ne se retrouvera donc jamais dans une situation où la contrainte de perte de paquets est vérifiée pour une solution non réalisable.

Les contraintes (5.6), (5.8) et (5.14) introduisent une non linéarité dans le modèle $B_{d,dp,k}$. Par ailleurs, ce problème a été démontré NP-difficile (Garey et Johnson, 1979). Nous allons maintenant présenter notre modèle concernant la perte de paquet $P_{d,p,k}$

5.2 Modèle avec contraintes de perte de paquets

Dans le modèle $P_{d,p,k}$, lorsqu'une requête arrive au contrôleur, celui-ci commence par calculer les probabilités potentielles qu'un paquet ne soit pas perdu ainsi que les délais pour chacun des liens du réseau. À partir des valeurs obtenues, le modèle est généré puis résolu. Finalement, la mise à jour des constantes est effectuée en fonction des résultats de la résolution du modèle. Ces opérations sont détaillées dans les sections suivantes.

5.2.1 Calcul des probabilités de perte de paquets et de réussite

Pour chaque lien $(i, j) \in M$, calculer

$$\overline{F_{ij}} = (\sum_{t \in T} \alpha^t y_{ij}^t) + \alpha$$

$$\lambda = \frac{\overline{F_{ij}}}{l_p}, \quad \rho = \frac{\overline{F_{ij}}}{C_{ij}}, \quad \overline{D_{ij}} = \frac{\rho[1 + k\rho^{k+1} - (k+1)\rho^k]}{\lambda(1-\rho)(1-\rho^k)}$$

$$\overline{P_{ij}} = \frac{\rho^k(1-\rho)}{1-\rho^{k+1}}, \quad \overline{R_{ij}} = 1 - \overline{P_{ij}}$$

Pour chaque LSP $(a,b) \in L$, calculer

$$\overline{D_{ab}} = \sum_{(i,j) \in M} \overline{D_{ij}} z_{ij}^{ab}$$

$$\overline{R_{ab}} = \prod_{(i,j) \in M : z_{ij}^{ab} = 1} \overline{R_{ij}}$$

Pour chaque chemin $c \in C_2$, calculer

$$\overline{R_c} = \prod_{(a,b) \in c} \overline{R_{ab}}.$$

Nous calculons aussi le délai car cette expression nous sert dans certains cas pour l'objectif.

5.2.2 Génération du modèle

Modèle $P_{d,p,k}$

$P_{d,p,k}$:

$$\min_{\{x_{ab} : (a,b) \in L\}} \sum_{(a,b) \in L} \overline{D_{ab}} x_{ab} \tag{5.15}$$

sujet à

$$y_{ij} = \sum_{(a,b) \in L} z_{ij}^{ab} x_{ab} \quad \forall (i,j) \in M \tag{5.16}$$

$$0 \le y_{ij} \le 1 \quad \forall (i,j) \in M \tag{5.17}$$

$$\sum_{(a,b) \in L} x_{ab} - \sum_{(a,b) \in L} x_{ba} \begin{cases} = 1 & \text{si b = destination} \\ = -1 & \text{si b = origine} \\ = 0 & \text{pour les autres cas} \end{cases} \tag{5.18}$$

$$x_{ab} \in \{0,1\} \quad \forall\,(a,b) \in L \tag{5.19}$$

$$\sum_{(a,b)\in L} x_{ab} \leq s_{\max} \tag{5.20}$$

$$\sum_{(a,b)\in L} (x_{ab} \ln \overline{R_{ab}}) \geq \ln(1-\phi) \tag{5.21}$$

$$\sum_{(i,j)\in M} \left\{ \ln(R_{ij}) + y_{ij} \ln\left(\frac{\overline{R_{ij}}}{R_{ij}}\right) \right\} y_{ij}^{t} \geq \ln(1-\phi^{t}) \quad (\forall t \in T) \tag{5.22}$$

L'objectif (5.15) reste le même que celui du modèle de base. Les contraintes (5.16) à (5.20) jouent respectivement le même rôle que les contraintes (4.14) et (4.18). La contrainte (5.21) permet de respecter le taux de perte maximal de la nouvelle connexion tandis que la perte de paquets pour les connexions déjà en service est assurée par les contraintes (5.22). Les contraintes (5.21) et (5.22) sont expliquées dans les deux paragraphes suivants. Les variables y_{ij} sont toujours entières. Par ailleurs, les contraintes de capacités sont omises car un dépassement de capacité se traduit par un faible taux de succès et les chemins avec un taux de succès inférieur au taux de succès minimal de la connexion ne seront automatiquement pas retenus (contraintes (5.21)).

Réduction du nombre de contraintes

Comme à la section 4.1.6.2, nous pouvons utiliser une approche par chemin car chaque chemin comprendra au maximum 2 LSP. Nous créons un tableau qui conserve le maximum des minima de probabilités de succès autorisés pour les connexions empruntant un seul LSP ou deux LSP consécutifs ($R_c^{\min}$). En bâtissant notre modèle, on calcule pour chaque LSP et pour chaque paire de LSP consécutifs le taux de succès minimum potentiel $\overline{R_c} = \prod_{(a,b)\in c} \overline{R_{ab}} \quad \forall c \in C_2$. Si ce produit est inférieur au minimum autorisé ($\overline{R_c} \leq R_c^{\min}$), la contrainte est écrite pour le chemin car il y a un risque potentiel de dépassement. Dans le cas contraire, il ne sert à rien d'écrire une contrainte pour ce chemin car, dans le pire cas, le taux de succès sera toujours supérieur au taux minimal

autorisé. Cela permet de réduire le nombre de contraintes à écrire. Les contraintes (5.22) pourront être remplacées par les contraintes (5.23) :

$$\sum_{(a,b)\in c}\sum_{(i,j)\in M}\left\{\ln(R_{ij})+y_{ij}\ln(\frac{\overline{R_{ij}}}{R_{ij}})\right\}z_{ij}^{ab}\geq\ln(R_c^{\min}) \qquad (5.23)$$
$$\forall c\in C_2,\ \overline{R_c}\leq R_c^{\min}$$

Les contraintes (5.23) permettent donc de respecter les contraintes de perte de paquets pour chaque chemin composé d'un ou deux LSP. Bien que toujours NP-difficile, ce modèle est linéaire et le faible nombre de variables permet de le résoudre rapidement, ce qui sera démontrée dans les sections suivantes.

Explication des contraintes (5.21) et (5.23)

Contraintes (5.21)

Taux de perte du chemin ≤ Taux de perte maximal de la connexion

Taux de succès du chemin ≥ Taux de succès minimal de la connexion

$$\prod_{(a,b)\in L}(1-\overline{P_{ab}}x_{ab})\geq(1-\phi)$$

x_{ab} étant une variable 0-1, deux cas peuvent se produire pour le membre de gauche. Si $x_{ab}=0$, le membre de gauche de l'inéquation vaut 1, élément neutre de la multiplication. Si $x_{ab}=1$, le membre de gauche de l'inéquation vaut $1-\overline{P_{ab}}=\overline{R_{ab}}$. Or, à cause des contraintes (5.18) de conservation de flot, pour toute solution réalisable, au moins une des variables x_{ab} doit valoir 1. On peut donc écrire :

$$\prod_{(a,b)\in L}(1-\overline{P_{ab}}x_{ab})\geq(1-\phi)\Leftrightarrow\prod_{(a,b)\in L/x_{ab}=1}(\overline{R_{ab}}x_{ab})\geq 1-\phi$$

A partir de cela, nous obtenons l'expression (5.21) :

$$\prod_{(a,b)\in L/x_{ab}=1}(\overline{R_{ab}}x_{ab})\geq 1-\phi$$
$$\ln(\prod_{(a,b)\in L/x_{ab}=1}\overline{R_{ab}}x_{ab})\geq\ln(1-\phi)$$
$$\sum_{(a,b)\in L/x_{ab}=1}\ln(\overline{R_{ab}}x_{ab})\geq\ln(1-\phi)$$

Puisque x_{ab} est une variable 0-1, on peut écrire pour généraliser et linéariser:

$$\sum_{(a,b)\in L}(x_{ab}\ln\overline{R_{ab}}) \geq \ln(1-\phi)$$

Ainsi, l'impact de $\overline{R_{ab}}$ ne sera pris en compte que si $x_{ab}=1$ (le LSP est utilisé).

Contraintes (5.23)

Cette contrainte concerne la perte de paquets pour les connexions en service. Prenons l'exemple de la Figure 5.1. Pour le LSP (i, l), on va calculer R_c et $\overline{R_c}$.

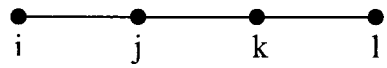

Figure 5.1 Chemin simple de i vers l

$$\begin{aligned} R_c &= R_{ij}\times R_{jk}\times R_{kl} \\ \overline{R_c} &= \overline{R_{ij}}\times\overline{R_{jk}}\times\overline{R_{kl}} \end{aligned} \qquad (5.24)$$

Notre objectif est que le résultat de la multiplication des taux de succès des différents liens soit supérieur au taux de succès minimal du chemin. Or il est possible que la nouvelle connexion n'emprunte pas tous les liens du LSP (i, l). Lorsqu'un lien est utilisé, il faut tenir compte de la valeur $\overline{R_{ij}}$ sinon c'est la valeur R_{ij} qui est utilisé. Considérons la relation suivante :

$$\prod_{(u,v)\in(i,l)} R_{uv}\times\prod_{(u,v)\in(i,l):y_{uv}=1}\frac{\overline{R_{uv}}}{R_{uv}}y_{uv} \geq R_c^{\min} \qquad (5.25)$$

Dans cette expression, si tous les liens du LSP (i, l) sont utilisés ($y_{ij}=y_{jk}=y_{kl}=1$), on se retrouve avec l'expression de $\overline{R_c}$ qui représente le pire cas. Comme exemple, développons cette inéquation en considérant que seul le lien (j, k) est utilisé par la nouvelle connexion.

$$\begin{aligned} &R_{ij}\times R_{jk}\times R_{kl}\times\frac{\overline{R_{jk}}}{R_{jk}}y_{jk} \geq R_c^{\min} \\ &R_{ij}\times\overline{R_{jk}}\times R_{kl} \geq R_c^{\min} \end{aligned} \qquad (5.26)$$

On se rend compte que si un lien n'est pas utilisé, son impact est R_{ij} tandis que s'il est utilisé, son impact est $\overline{R_{ij}}$. Faisons un développement logarithmique de l'expression (5.25).

$$\begin{aligned}
&\ln(\prod_{(u,v)\in(i,l)} R_{uv} \times \prod_{(u,v)\in(i,l):y_{uv}=1} \frac{\overline{R_{uv}}}{R_{uv}} y_{uv}) \geq \ln(R_c^{\min}) \\
&\sum_{(u,v)\in(i,l)} \ln R_{uv} + \sum_{(u,v)\in(i,l):y_{uv}=1} \ln(\frac{\overline{R_{uv}}}{R_{uv}} y_{uv}) \geq \ln(R_c^{\min}) \\
&\sum_{(u,v)\in(i,l)} \ln R_{uv} + \sum_{(u,v)\in(i,l)} y_{uv} \ln \frac{\overline{R_{uv}}}{R_{uv}} \geq \ln(R_c^{\min}) \\
&\sum_{(u,v)\in(i,l)} (\ln R_{uv} + y_{uv} \ln \frac{\overline{R_{uv}}}{R_{uv}}) \geq \ln(R_c^{\min})
\end{aligned} \tag{5.27}$$

Si le chemin est composé de plusieurs LSP, on retombe sur l'expression 5.23.

$$\sum_{(a,b)\in c} \sum_{(i,j)\in M} \left\{ \ln(R_{ij}) + y_{ij} \ln(\frac{\overline{R_{ij}}}{R_{ij}}) \right\} z_{ij}^{ab} \geq \ln(R_c^{\min})$$

$$\forall c \in C_2, \overline{R_c} \leq R_c^{\min}$$

Mentionnons que la plupart des applications requérant de la QdS nécessitent un taux de perte inférieur à 10% (donc un taux de réussite supérieur à 90%) et on aura $1 \geq R_{ij} \geq 0.9$. Comme la fonction du calcul de la perte de paquets tend vers 1 lorsque le flot tend vers l'infini (équation (4.2)), on peut dire que le taux de succès tend vers 0 à l'infini, ce qui nous permet d'écrire $1 > \overline{R_{ij}} > 0$. L'expression $\ln(\frac{\overline{R_{ij}}}{R_{ij}})$ sera donc toujours calculable. Ainsi, avec l'expression développée (5.23), nous arrivons à imposer la contrainte pour garantir un niveau de succès dans la livraison des paquets.

Ce modèle est linéaire et cela permet une résolution rapide par les solveurs.

5.2.3 Mise à jour

La nouvelle connexion est désignée par $t' \in T$. Voici les opérations à effectuer seulement si la connexion est acceptée :

Pour chaque lien $(i, j) \in M$,

Si la nouvelle connexion utilise le lien $(i, j) \in M$, alors

$$y_{ij}^{t'} = 1$$

Sinon

$$y_{ij}^{t'} = 0$$

Pour le chemin $c \in C$ utilisé par $t' \in T$,

Si $\phi^{t'} < P_c^{\max}$, alors

$$P_c^{\max} = \phi^{t'}$$

$$R_c^{\min} = 1 - P_c^{\max}$$

5.2.4 Objectif de perte de paquets

Lorsque nous voulons écrire le modèle en considérant la perte de paquets comme objectif de minimisation, nous allons plutôt tenter de maximiser la probabilité de succès sur un chemin. La fonction suivante est utilisée :

$$\max_{\{x_{ab}:(a,b)\in L\}} \sum_{(a,b)\in L} (x_{ab} \ln \overline{R_{ab}}) \qquad (5.28)$$

Cette expression est extraite des contraintes (5.21).

5.3 Modèle multi-contraintes

Après avoir présenté les modèles pour le délai et la perte de paquets, nous allons maintenant considérer le problème multi-contraintes $P_{d,dp,k}$ qui est plus complexe à résoudre. Ce modèle combine les contraintes des modèles $P_{d,d,k}$ et $P_{d,p,k}$. Les étapes de résolutions sont les mêmes que celles utilisées pour le modèle précédent.

5.3.1 Calcul des délais, des probabilités de perte et de réussite

Pour chaque lien $(i, j) \in M$, calculer

$$\overline{F_{ij}} = (\sum_{t \in T} \alpha^t y_{ij}^t) + \alpha$$

$$\lambda = \frac{\overline{F_{ij}}}{l_p}, \quad \rho = \frac{\overline{F_{ij}}}{C_{ij}}, \quad \overline{D_{ij}} = \frac{\rho[1 + k\rho^{k+1} - (k+1)\rho^k]}{\lambda(1-\rho)(1-\rho^k)}$$

$$\overline{P_{ij}} = \frac{\rho^k(1-\rho)}{1-\rho^{k+1}}, \quad \overline{R_{ij}} = 1 - \overline{P_{ij}}$$

$$\Delta D_{ij} = \overline{D_{ij}} - D_{ij}$$

Pour chaque LSP $(a,b) \in L$, calculer

$$\overline{D_{ab}} = \sum_{(i,j) \in M} \overline{D_{ij}} z_{ij}^{ab}$$

$$\overline{R_{ab}} = \prod_{(i,j) \in M : z_{ij}^{ab} = 1} \overline{R_{ij}}$$

Pour chaque chemin $c \in C_2$, calculer

$$\overline{D_c} = \sum_{(a,b) \in c} \overline{D_{ab}}$$

$$\overline{R_c} = \prod_{(a,b) \in c} \overline{R_{ab}}.$$

5.3.2 Génération du modèle

Modèle $P_{d,dp,k}$

$P_{d,dp,k}$:

$$\min_{\{x_{ab} : (a,b) \in L\}} \sum_{(a,b) \in L} \overline{D_{ab}} x_{ab} \tag{5.29}$$

sujet à

$$y_{ij} = \sum_{(a,b) \in L} z_{ij}^{ab} x_{ab} \quad \forall (i,j) \in M \tag{5.30}$$

$$0 \le y_{ij} \le 1 \quad \forall (i,j) \in M \tag{5.31}$$

$$\sum_{(a,b) \in L} x_{ab} - \sum_{(a,b) \in L} x_{ba} \begin{cases} = 1 & \text{si b = destination} \\ = -1 & \text{si b = origine} \\ = 0 & \text{pour les autres cas} \end{cases} \tag{5.32}$$

$$x_{ab} \in \{0,1\} \quad \forall (a,b) \in L \tag{5.33}$$

$$\sum_{(a,b)\in L} x_{ab} \leq s_{\max} \qquad (5.34) \tag{5.34}$$

$$\sum_{(a,b)\in L} \overline{D_{ab}} x_{ab} \leq \beta \tag{5.35}$$

$$\sum_{(a,b)\in c} \sum_{(i,j)\in M} (D_{ij} + y_{ij}\Delta D_{ij}) z_{ij}^{ab} \leq D_c^{\max} \quad \forall c \in C_2, \overline{D_c} \geq D_c^{\max} \tag{5.36}$$

$$\sum_{(a,b)\in L} (x_{ab} \ln \overline{R_{ab}}) \geq \ln(1-\phi) \tag{5.37}$$

$$\sum_{(a,b)\in c} \sum_{(i,j)\in M} \left\{ \ln(R_{ij}) + y_{ij} \ln(\frac{\overline{R_{ij}}}{R_{ij}}) \right\} z_{ij}^{ab} \geq \ln(R_c^{\min}) \quad \forall c \in C_2, \overline{R_c} \leq R_c^{\min} \tag{5.38}$$

Les contraintes (5.29) à (5.34) sont les mêmes que les contraintes (5.15) à (5.20). Les contraintes de délais (5.35) et (5.36) sont identiques aux contraintes (4.19) et (4.21) tandis que les contraintes de probabilité de pertes de paquets (5.37) et (5.38) sont similaires à celles représentées par les contraintes (5.21) et (5.23). Ce modèle devrait être un peu plus long à écrire car les contraintes de QdS sont plus nombreuses mais le nombre de variables ne change pas.

5.3.3 Mise à jour

En s'inspirant des modèles précédents et en désignant par $t' \in T$ la nouvelle connexion, voici les opérations à effectuer seulement si la connexion est acceptée :

Pour chaque lien $(i, j) \in M$,

Si la nouvelle connexion utilise le lien $(i, j) \in M$, alors

$y_{ij}^{t'} = 1$

Sinon

$y_{ij}^{t'} = 0$

Pour le chemin $c \in C$ utilisé par $t' \in T$,

Si $\phi^{t'} < P_c^{\max}$, alors

$$P_c^{\max} = \phi^{t'}$$

$$R_c^{\min} = 1 - P_c^{\max}$$

Si $\beta^{t'} < D_c^{\max}$, alors

$$D_c^{\max} = \beta^{t'}$$

Après avoir présenté nos différents modèles, nous allons passer maintenant à l'évaluation de performance.

5.4 Évaluation de performance

Nous utiliserons les mêmes jeux de tests et les métriques décrites aux sections 4.2.1 et 4.2.2. L'environnement de test est le même que celui décrit en 4.2.1. Dans cette section, nous allons d'abord étudier les performances du modèle P avec les contraintes de perte de paquets puis celles du modèle mixte. Pour des raisons de simplicité de présentation, nous considérons $k = 800$ paquets qui est une valeur raisonnable pour la taille de la file.

5.4.1 Contraintes de perte de paquets

Nous allons comparer les modèles ayant pour objectif de minimiser le délai $P_{d,p,k}$ et la perte de paquets $P_{p,p,k}$ respectivement avec les modèles de références $R_{d,p,k}$ et $R_{p,p,k}$.

5.4.1.1 Impact du nombre maximal de LSP autorisés

Les Figures (5.2a) et (5.2b) présentent le taux de blocage en fonction de S_{max} pour les connexions avec l'utilisation de files M/M/1/k pour $P_{d,p,k}$ et $P_{p,p,k}$. Les résultats pour le modèle avec objectif de délai sont similaires. Pour le modèle avec objectif de pertes de paquets, on se rend compte qu'en moyenne, plus il y a de LSP autorisés, plus le blocage est important.

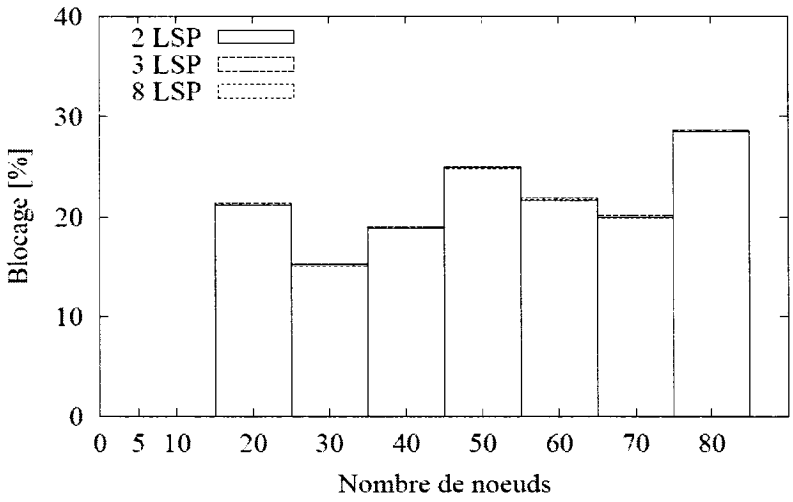

Figure 5.2a Blocage ($P_{d,p,k}$)

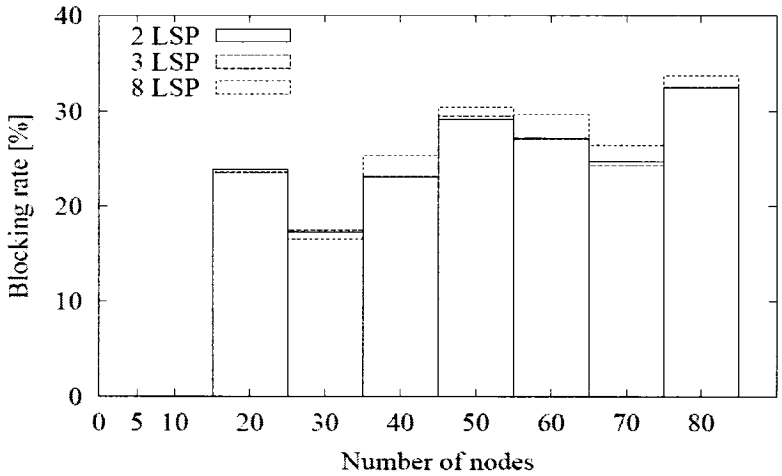

Figure 5.2b Blocage ($P_{p,p,k}$)

Une explication peut être donnée à ce constat étonnant. En effet, par rapport à la formule (4.2), la perte de paquets est quasiment nulle jusqu'à la zone de saturation ce qui veut dire que la probabilité de succès vaut 1 avant la zone de saturation. Puisque l'objectif du modèle $P_{p,p,k}$ est l'expression $\max_{\{x_{ab}:(a,b)\in L\}} \sum_{(a,b)\in L} (x_{ab} \ln \overline{R_{ab}})$, cet objectif est toujours nul avant la zone de saturation des liens, ce qui veut dire que, jusqu'à un certain point, les solutions sont choisies aléatoirement dans l'espace des solutions réalisables. De

ce fait, certains chemins utilisant un grand nombre de LSP peuvent être choisis, ce qui entraîne une surcharge inutile du réseau. Plus le nombre de LSP autorisé est grand, plus vite les liens peuvent arriver à saturation et augmenter le blocage. Cette explication est valable pour tous les autres résultats obtenus avec les modèles ayant pour objectif la minimisation de la probabilité de perte de paquets.

5.4.1.2 Blocage des connexions

Les Figures (5.3a) et (5.3b) traitent du blocage des connexions pour les différents objectifs étudiés.

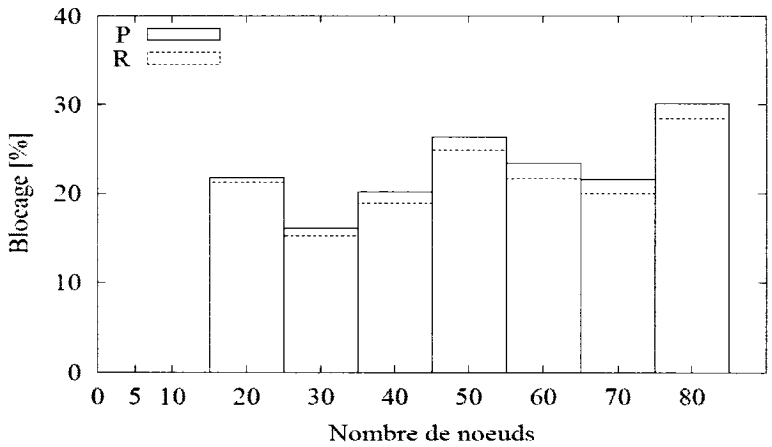

Figure 5.3a Blocage ($P_{d,p,k}$)

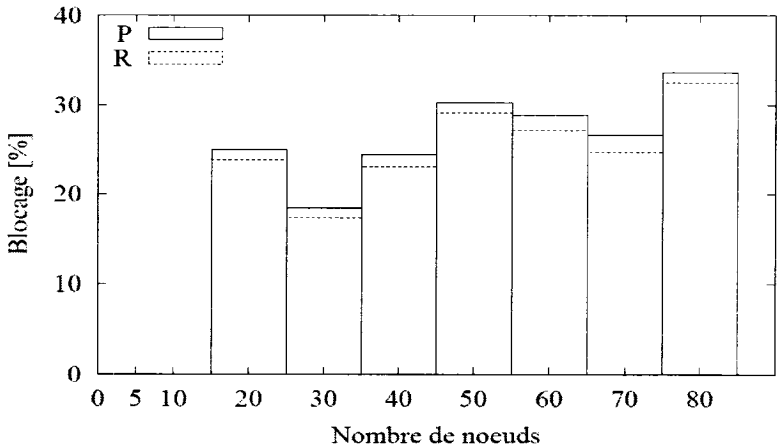

Figure 5.3b Blocage ($P_{p,p,k}$)

On remarque que l'écart par rapport aux modèles de références est inférieur à 3% quelle que soit la fonction objectif. On constate aussi que le modèle $P_{p,p,k}$ occasionne plus

de blocage en moyenne que le modèle $P_{d,p,k}$. On se réfèrera aux explications données dans la section 5.4.1.1 pour justifier cet état de fait.

5.4.1.3 Probabilité de perte de paquets de bout en bout

Les modèles proposés permettent de conserver la probabilité de perte de paquets en dessous de 0.4% quel que soit l'objectif utilisé.

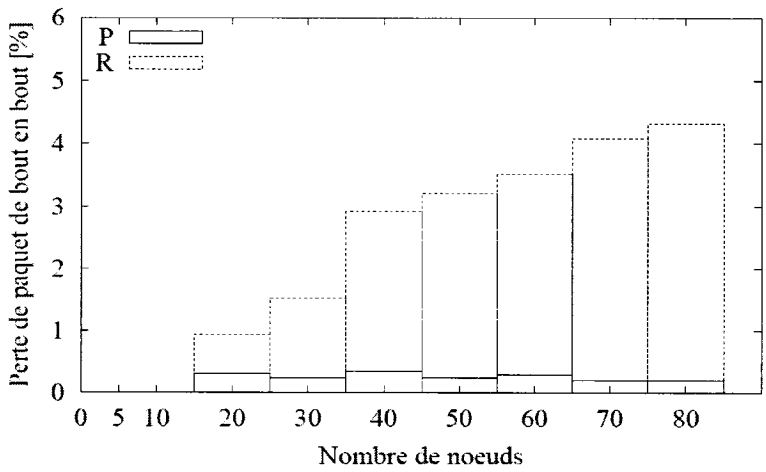

Figure 5.4a Probabilité de perte de paquets de bout en bout ($P_{d,p,k}$)

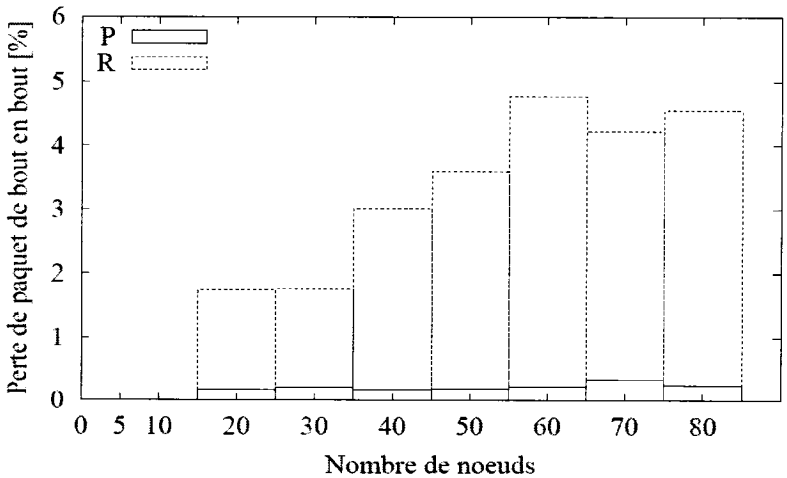

Figure 5.4b Probabilité de perte de paquets de bout en bout ($P_{p,p,k}$)

Dans le modèle $P_{p,p,k}$, même si comme nous l'avons expliqué l'objectif peut être nul dans certains cas, le choix de la solution pour la nouvelle connexion se fait néanmoins en

tenant compte de toutes les autres connexions. De ce fait, la limite de probabilité de perte de paquet est respectée pour chaque connexion.

5.4.1.4 Ratio de connexions ayant dépassées leur perte de paquets maximale

Les deux modèles de référence procurent des ratios de dépassement atteignant 70% au maximum.

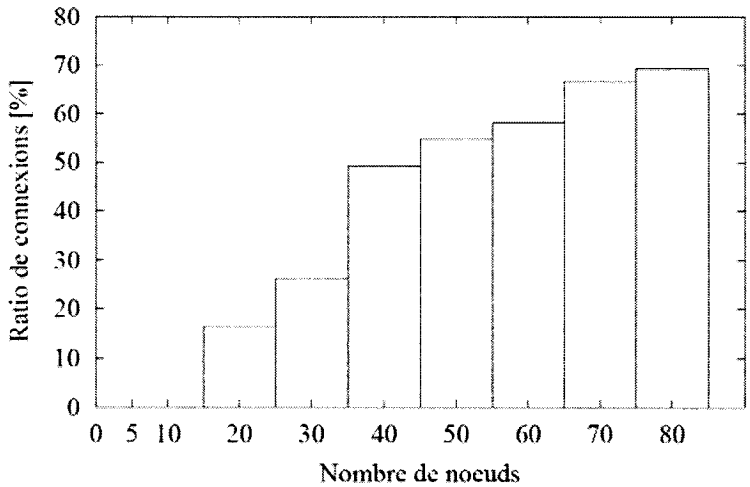

Figure 5.5a Ratio de dépassement de la probabilité de perte de paquets ($R_{d,p,k}$)

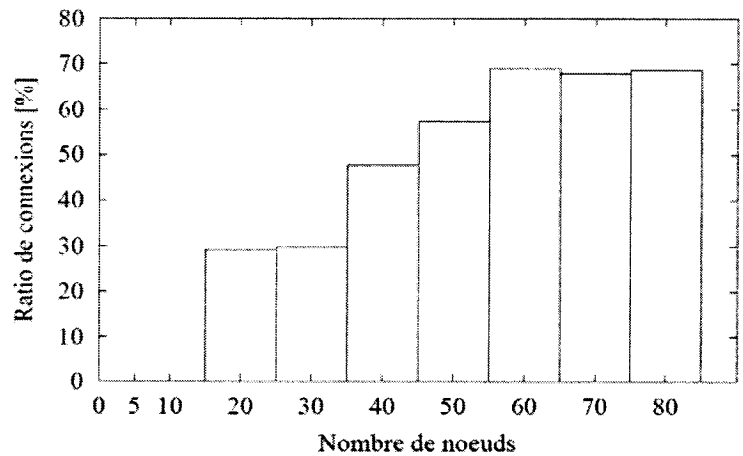

Figure 5.5b Ratio de dépassement de la probabilité de perte de paquets ($R_{p,p,k}$)

Nous constatons que le modèle $R_{p,p,k}$ offre des ratios en moyenne supérieurs à ceux du modèle $R_{d,p,k}$, ce qui est logique puisque la perte de paquets moyenne de $R_{p,p,k}$ est supérieure à celle de $R_{d,p,k}$.

5.4.1.5 Temps d'exécution

Les Figures 5.6a et 5.6b présentent les temps d'exécution pour les modèles $P_{d,p,k}$ et $P_{p,p,k}$.

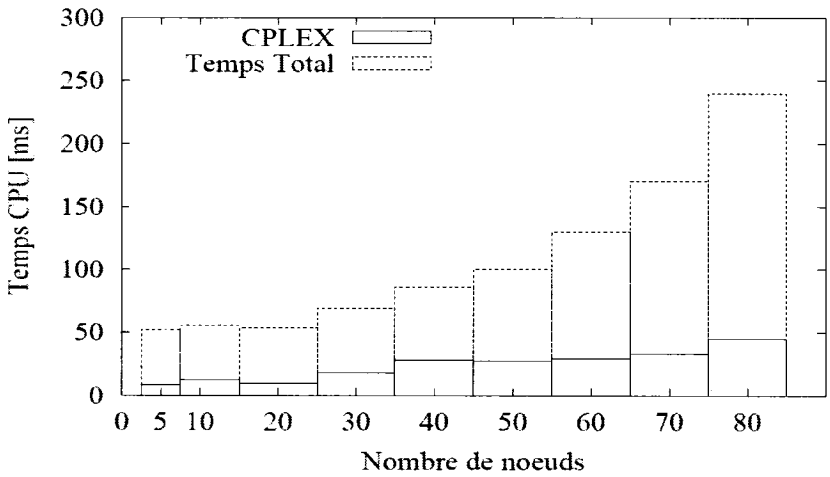

Figure 5.6a Temps d'exécution ($P_{d,p,k}$)

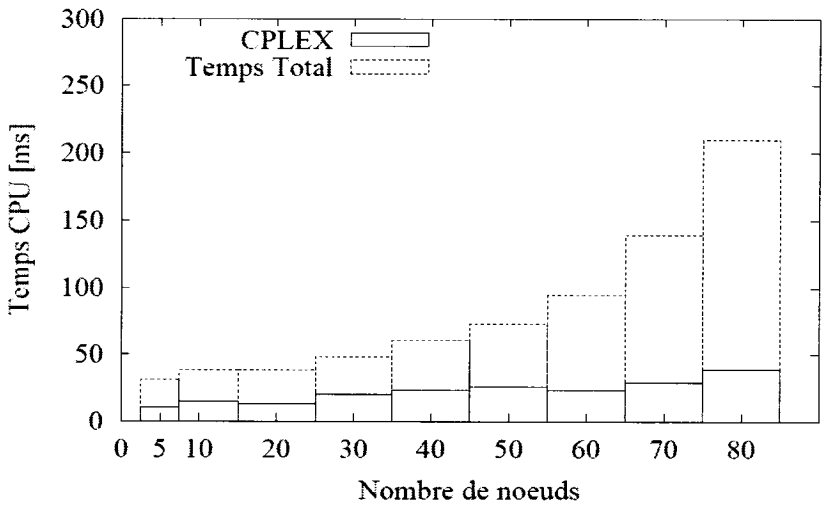

Figure 5.6b Temps d'exécution ($P_{p,p,k}$)

Les temps de résolution des modèles avec CPLEX sont en dessous de 50 ms. Le temps d'exécution total de $P_{d,p,k}$ atteint au maximum 245 ms tandis que celui de $P_{p,p,k}$ a un maximum de 210 ms. Cette observation est à mettre en relation avec le taux de blocage. Comme il y a plus de connexions bloquées avec $P_{p,p,k}$, la génération du modèle est moins longue à chaque itération.

5.4.1.6 Conclusion

Les deux modèles proposés fournissent des résultats acceptables dans des temps de calculs raisonnables. Toutefois, nous privilégions le modèle $P_{d,p,k}$ à cause du blocage moins important qu'il implique.

5.4.2 Problème multi-contraintes avec objectif de minimisation de délai

Cette section étudie le problème multi-contraintes ($P_{d,dp,k}$). Nous allons comparer ce modèle au modèle de référence ($R_{d,dp,k}$).

5.4.2.1 Impact du nombre de LSP autorisés

Nous observons sur la Figure (5.7) que le taux de blocage, en fonction de S_{max}, est similaire pour les connexions avec l'utilisation de files M/M/1/k pour $P_{d,dp,k}$. Le fait de limiter le nombre maximum de LSP utilisés pour le routage des connexions n'a pas d'impact significatif sur le taux de blocage.

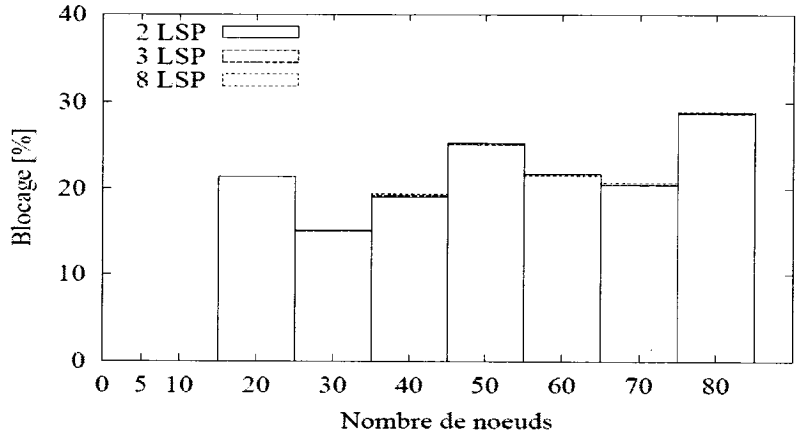

Figure 5.7 Blocage ($P_{d,dp,k}$)

5.4.2.2 Blocage des connexions

La différence maximale au niveau du taux de blocage est de 7%. Cette différence est majoritairement attribuable aux contraintes de perte de paquets car le délai de la file est borné à 100 ms (Figure 4.2) ce qui permettrait d'accepter certaines connexions largement au delà d'une utilisation du lien de 100%. Par contre, la perte de paquets sur un lien atteint rapidement la limite maximale de 5% que nous avons fixés. De ce fait, les contraintes de pertes de paquets auront plus d'impact au niveau du blocage des connexions.

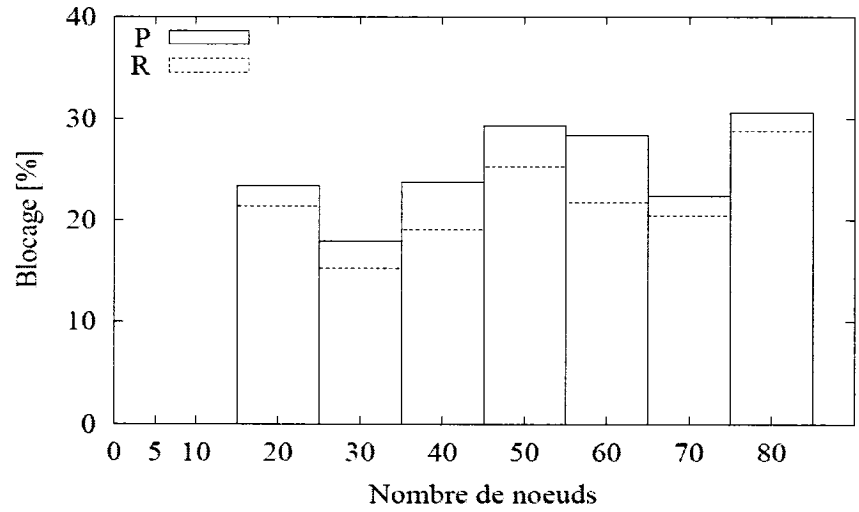

Figure 5.8 Blocage ($P_{d,dp,k}$)

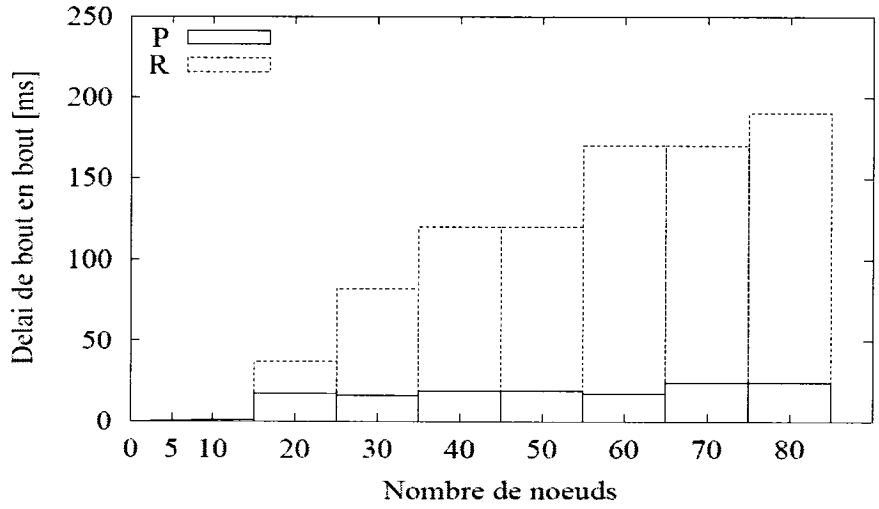

Figure 5.9a Délai de bout en bout

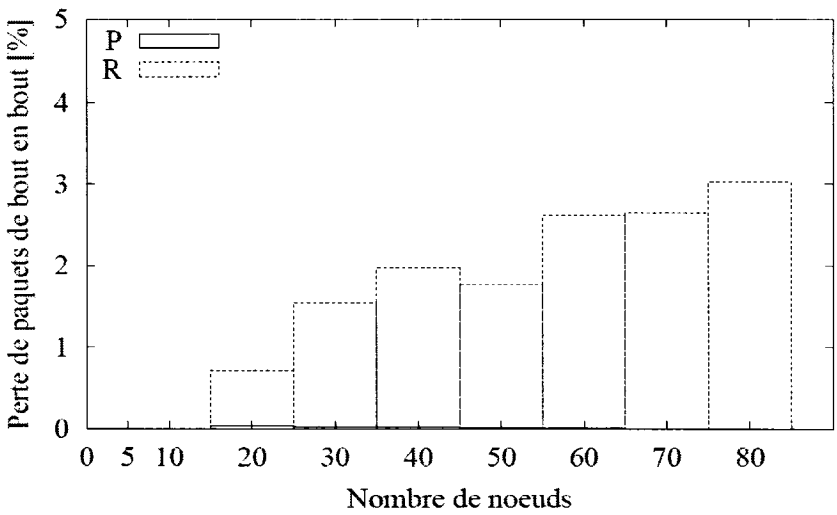

Figure 5.9b Probabilité de perte de paquets de bout en bout

5.4.2.3 Délai et probabilité de perte de paquets de bout en bout

Les résultats montrent que *P* conserve les délais moyens en dessous de 25 ms quand *R* atteint des délais de 190 ms. Pour la perte de paquets, *P* fournit des valeurs quasiment nulles alors que *R* atteint les 3%. Ces résultats ne sont pas surprenants à cause des contraintes que nous rajoutons.

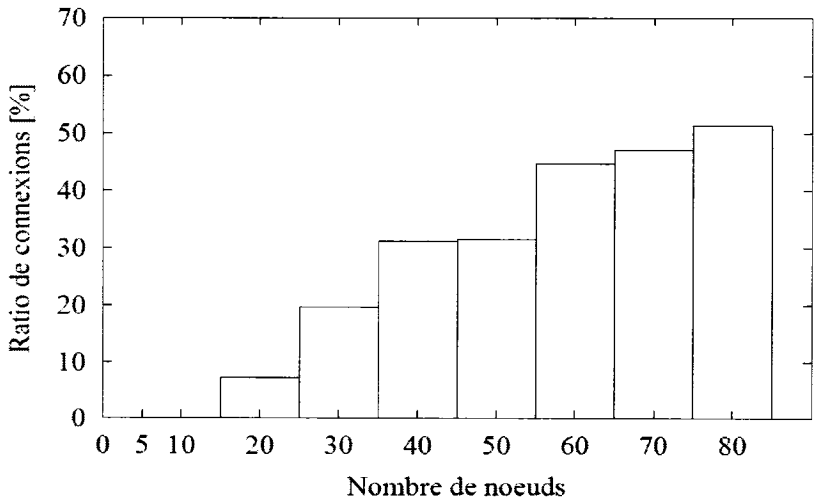

Figure 5.10a Ratio de connexions en dépassement - Délai

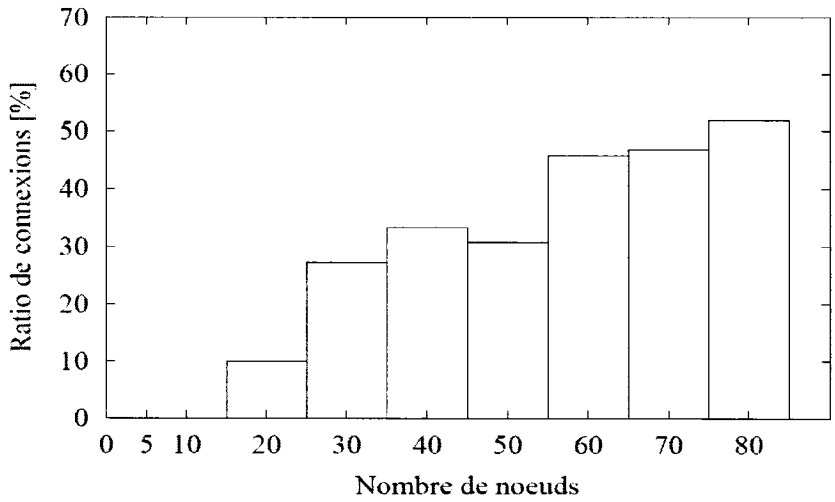

Figure 5.10b Ratio de connexions en dépassement - Perte de paquets

5.4.2.4 Ratio de connexions ayant dépassé leurs limites

Sur les Figures (5.10a) et (5.10b), nous constatons que, sans l'ajout des nouvelles contraintes, il y aurait un maximum de 51% de connexions qui auraient leur délai dépassé et un maximum de 53% qui serait en dépassement de la limite permise de perte de paquets. Ces résultats confirment l'apport non négligeable de nos propositions.

5.4.2.5 Temps d'exécution

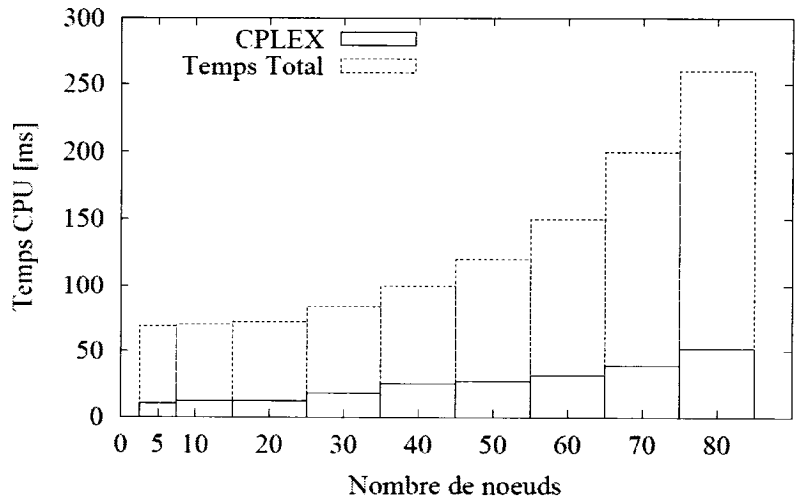

Figure 5.11 Temps d'exécution

Comme dans tous les autres cas, la résolution du modèle avec le logiciel CPLEX ne prend pas plus de 53 ms tandis que le temps total de résolution ne dépasse pas 260 ms, ce qui est très acceptable dans un cadre de contrôle d'admission dynamique. Au regard de tous ces résultats, nous pouvons affirmer que le modèle $P_{d,dp,k}$ est une excellente alternative au problème de CAC multi-contraintes. Nous allons pousser notre analyse pour étudier les performances avec différentes fonctions objectifs.

5.4.3 Problème multi-contraintes avec différents objectifs

Cette section étudie le problème multi-contraintes avec différents objectifs de minimisation. Nous considérons l'objectif de délai $P_{d,dp,k}$, l'objectif de perte de paquets $P_{p,dp,k}$ et l'objectif à coûts fixes $P_{f,dp,k}$. Pour des raisons de simplicité sur les figures de cette section, *Pd*, *Pp* et *Pf* réfèreront respectivement aux objectifs de délai, de perte de paquets et de coûts fixes. Nous ne présenterons pas la figure concernant les ratios de dépassement car ils sont tous nuls pour les trois modèles étudiés à cause des nouvelles contraintes ajoutées pour toutes les connexions du réseau.

5.4.3.1 Blocage des connexions

La Figure 5.12 montre que le modèle $P_{d,dp,k}$ fournit les meilleurs résultats en termes de blocage (maximum de 30%) tandis que le modèle $P_{f,dp,k}$ nous donne les moins bonnes valeurs (maximum de 45%). Le modèle $P_{p,dp,k}$ se situe entre les deux et permet d'atteindre un maximum de 36%. L'écart entre le meilleur et le pire cas atteint 18% pour le réseau de 70 nœuds. L'objectif de minimisation de la perte de paquets donne plus de blocage à cause de la nature de la fonction objectif. En effet, avant d'atteindre un certain niveau de charge du réseau, le choix des chemins se fait de façon aléatoire car les probabilités de succès valent 1. $P_{f,dp,k}$ donne les pires résultats car les coûts des liens sont fixes, ce qui diminue la tendance à balancer la charge sur le réseau. En effet, pour une paire origine destination donnée, l'ordre de sélection des chemins est toujours le même à cause des coûts fixes, ce qui entraîne une congestion plus rapide de certains liens et augmente le blocage.

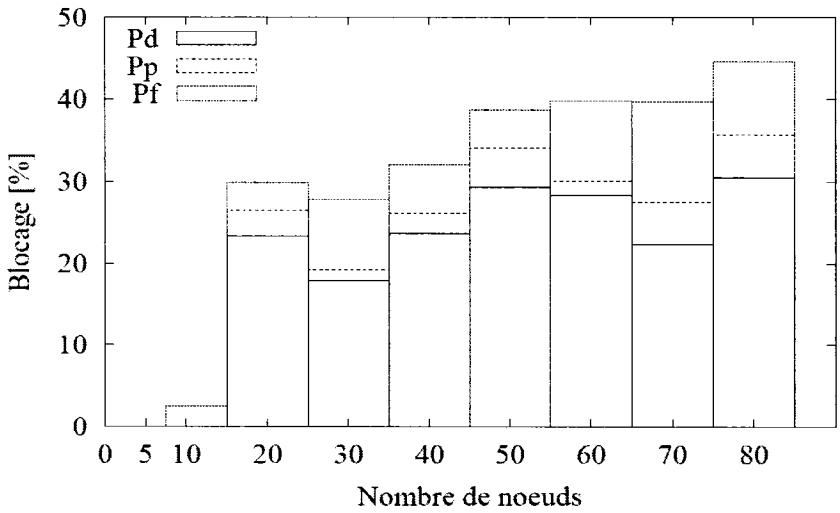

Figure 5.12 Blocage

5.4.3.2 Délai et probabilité de perte de paquets de bout en bout

La Figure 5.13a nous montre que les délais sont tous en dessous de 25 ms. Quelques fois, le modèle avec objectif de minimisation de délais donne de moins bons délais moyens que les autres. Cela est à mettre en relation avec le taux de blocage. Comme ce modèle permet d'accepter plus de connexions, il est normal que le délai moyen puisse être quelque fois supérieur aux autres modèles. Toutefois, les valeurs obtenues sont largement satisfaisantes.

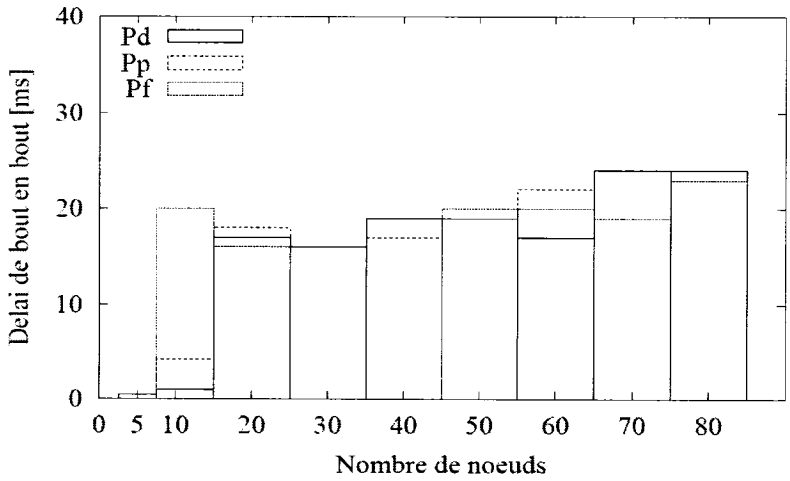

Figure 5.13a Délai de bout en bout

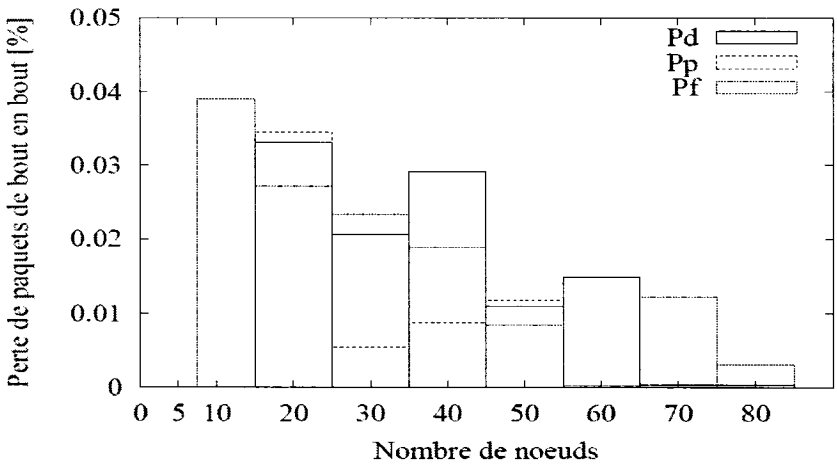

Figure 5.13b Probabilité de perte de paquets de bout en bout

Par rapport à la perte de paquets (Figure 5.13b), une tendance nette ne s'observe pas mais puisque tous les résultats sont en dessous de 0.04%, nous jugeons tous les modèles satisfaisants sur ce point.

5.4.3.3 Temps d'exécution

Au niveau des temps d'exécution, le modèle $P_{d,dp,k}$ qui nécessite le plus de temps, se résout en moins de 53 ms par le logiciel CPLEX et l'écart maximal entre les solutions des différents modèles ne dépasse pas 5 ms (Figure 5.14a).

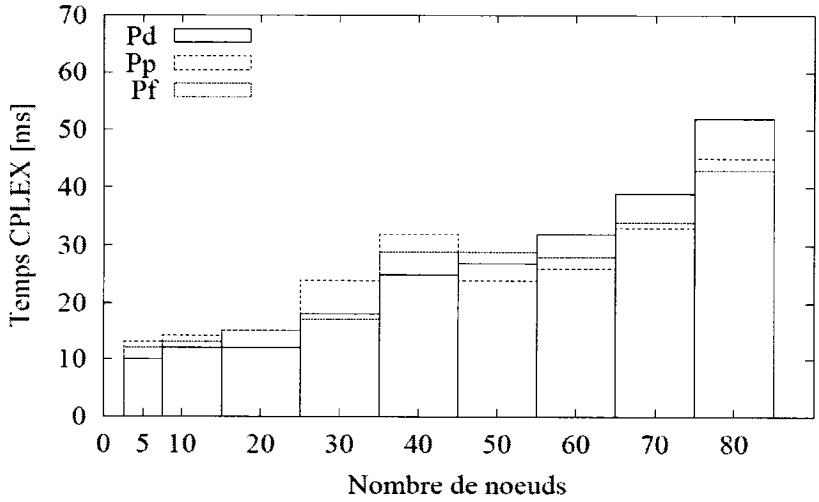

Figure 5.14a Temps CPLEX

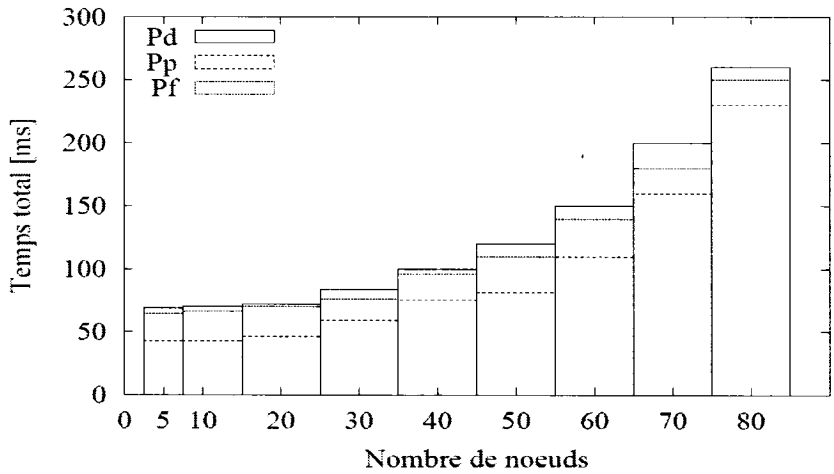

Figure 5.14b Temps total d'exécution

Pour le temps total, il faut moins de 260 ms au modèle $P_{d,dp,k}$ dans tous les cas et l'écart maximal entre les solutions des différents modèles ne dépasse pas 40 ms (Figure 5.14b). Le fait que le modèle $P_{d,dp,k}$ nécessite plus de temps est intimement lié au taux de blocage moins élevé qu'il fournit. Plus il y a de connexions acceptées, plus le temps de résolution sera élevé. Toutefois, les valeurs obtenues permettent de faire un contrôle d'admission dynamique.

5.4.3.4 Conclusions

Au regard de tous ces résultats, nous recommandons le modèle multi-contraintes avec objectif de minimisation de délai. Au niveau du blocage des connexions, la différence par rapport aux autres modèles atteint 18% dans certains cas. En termes de délai et de probabilité de perte de paquets de bout en bout, les différents modèles fournissent des résultats largement acceptables et aucune différence notable n'a été décelée. Enfin, le temps de résolution global est acceptable puisque tous les modèles s'exécutent en moins de 260 ms et ce pour des réseaux de grandes tailles. Le modèle multi-contraintes avec objectif de minimisation de délai que nous préconisons nécessitent plus de temps pour les réseaux de grande taille (différence maximale de 40 ms) mais cela est intimement lié au fait que ce modèle permet l'acceptation d'un plus grand nombre de connexions.

CHAPITRE VI

CONCLUSION

L'évolution des réseaux de télécommunications tend vers la définition de réseaux de prochaines générations (NGN) capables d'offrir des services variés requérant des niveaux de QdS différents. Plusieurs organismes s'attèlent à la définition de telles architectures mais certaines pièces du puzzle restent à être insérées. Dans le vaste éventail de problèmes apportés par les NGN, nous avons choisi d'aborder les problématiques liées à l'utilisation des réseaux Ethernet Métropolitains et celle du contrôle d'admission dynamique des connexions dans les réseaux IP. Nous avons proposé des architectures et des mécanismes permettant d'atteindre ces objectifs.

6.1 Synthèse des travaux

Nous avons, dans un premier temps, fait une revue de la littérature pour présenter les NGN et différentes problématiques liées à leur déploiement. Parmi les propositions d'architectures pour les NGN, nous avons choisi de fonder notre étude sur l'architecture TISPAN car elle nous semble la plus complète à l'heure actuelle. Un aspect non encore défini de l'architecture TISPAN concerne le réseau d'agrégation de niveau 2. Nous avons proposé l'utilisation d'Ethernet comme technologie pour ce réseau d'agrégation à cause du rapport coût performance offert par les évolutions d'Ethernet.

Pour pouvoir utiliser Ethernet dans les NGN, nous avons abordé le problème de la QdS. Nous avons proposé une architecture de gestion de QdS avec un contrôleur d'admission centralisé. Cette architecture, inspirée de la proposition RMD (Bader *et al.*, 2006) pour les réseaux IP, permet d'assurer un certain niveau de QdS à chaque connexion tout en conservant une approche *DiffServ*. Les fonctions et les interactions entre le contrôleur et les nœuds frontières ont été définies. Par ailleurs, nous avons proposé une modification de l'entête Ethernet qui permet d'augmenter le nombre de

VLANs et de classes de service, ainsi que de définir des LSP tout en n'augmentant pas la taille des informations de contrôle, ce qui n'était pas possible auparavant. Cette architecture de QdS pour Ethernet a ensuite été insérée dans l'architecture globale de TISPAN et nous avons défini les interactions entre les gestionnaires des différents domaines de QdS. Les protocoles de réservations de ressources ont été validés formellement à l'aide du logiciel UPPAAL.

Une fois cette architecture définie, nous nous sommes attelés à la proposition d'heuristiques pour le contrôle d'admission dynamique des connexions (CAC). Nos heuristiques concernent trois types de problèmes : le problème avec contraintes de délai, le problème avec contraintes de pertes de paquets et le problème multi-contraintes. Ce sont des problèmes qui sont réputés difficiles à résoudre. Nous avons considéré un réseau MPLS avec une architecture logique de LSP mais nos solutions s'appliquent à tous types de réseaux utilisant des topologies logiques. Nos solutions trouvent le chemin qui va minimiser un paramètre donné (délai, perte de paquets ou coûts fixes) lorsque la nouvelle connexion sera en service, ce qui diffère de la plupart des solutions qui trouvent le plus court chemin sans considérer l'impact du nouveau flot.

De plus, la grande particularité de nos modèles est qu'ils considèrent les contraintes de QdS pour tous les flots en service. Nous assurons que les valeurs limites pour toutes les connexions ne sont pas dépassées. Les contraintes linéaires que nous avons introduites représentent un apport important à la modélisation du CAC. Nous évitons le reroutage des trafics déjà en service pour qu'il n'y ait pas d'interruption de service. Cet aspect est novateur car la plupart des travaux effectués sur la question ne considèrent pas les paramètres de délais et de perte de paquets de bout en bout pour les flots déjà en service. Nos modèles ont été testés à partir d'un code en langage C qui appelle séquentiellement les fonctions du logiciel CPLEX pour la résolution du modèle. Nous avons démontré que l'ajout des nouvelles contraintes a une influence limitée sur le taux de blocage (maximum de 7%) tout en réduisant le délai moyen et la perte de paquets. De plus, nos modèles empêchent le dépassement des limites de QdS fixées pour toutes les connexions. Sans les nouvelles contraintes, nous atteignons dans certains cas un ratio de dépassement de 70%.

On peut donc dire que bloquer l'accès a un maximum de 7 % de connexions pour permettre de satisfaire 70 % de connexions est un résultat largement acceptable.

En termes de temps d'exécution, la résolution de nos modèles ne prend pas plus de 53 ms quand l'exécution totale ne dépasse pas 260 ms, ce qui rend nos propositions applicables dans un cadre de CAC dynamique. Le temps de résolution des modèles que nous obtenons avec CPLEX est très satisfaisant. On observe que le nombre nœuds explorés par le *branch and bound* pour obtenir une solution optimale est très petit (maximum de 10). Cela peut s'expliquer par le fait que la relaxation linéaire fournit des solutions de bonne qualité mais aussi et surtout par le fait que nous effectuons plusieurs simplifications avant la résolution du modèle. Par exemple, lorsqu'un lien arrive dans la zone de saturation, son délai est mis à une valeur très grande. Ce faisant, tous les LSP passant par ce lien ne pourront faire partie de la solution à cause de la contrainte de délai. Nous réduisons ainsi l'espace des solutions réalisables. Toutefois, ces temps d'exécution pourraient être réduits en optimisant le code et en utilisant des machines plus puissantes, ce qui serait le cas pour des contrôleurs réels.

6.2 Limitations des travaux

Dans ce travail concernant la qualité de service des réseaux de prochaines générations, certaines limitations sont à mentionner. Premièrement, l'architecture TISPAN est en pleine définition. De ce fait, puisque nous avons défini une architecture Ethernet qui doit être intégrée au réseau d'accès TISPAN, l'interconnexion et les protocoles que nous avons définis sont sujets à changement en fonction de la standardisation de l'architecture TISPAN. Notre proposition pour Ethernet peut être vue comme une boîte noire qui va interagir avec le réseau TISPAN.

Par ailleurs, sur le plan architectural, nous n'avons pas introduit de nœud redondant pour notre contrôleur Ethernet. Or, puisque notre architecture est centralisée, une panne du contrôleur pourrait mettre hors service tout le réseau Ethernet. Un autre aspect qui dépendra de la standardisation de l'architecture TISPAN est la localisation physique de notre contrôleur Ethernet. En effet, nous avons choisi de séparer le contrôleur Ethernet et

le contrôleur déjà défini dans TISPAN pour permettre à des opérateurs différents de coopérer. Toutefois, les fonctions de contrôle Ethernet pourraientt aussi être effectuées par la même entité qui contrôle globalement le réseau d'accès, permettant ainsi à un seul opérateur de gérer tout le réseau d'accès.

D'autre part, nous n'avons pas pu implémenter nos solutions concernant Ethernet et TISPAN pour faire des simulations, car l'architecture TISPAN est encore dans ses stades primaires de définition et aucun logiciel n'implémente les fonctionnalités requises. Nous devons donc attendre la maturation de TISPAN pour pouvoir faire des simulations. En outre, les aspects de sécurité n'ont pas été abordés.

Concernant le contrôle d'admission des connexions, le fait de choisir un modèle de contrôle d'admission basé sur les réservations peut résulter en un faible taux d'utilisation du réseau. En effet, une connexion donnée peut ne pas utiliser toute la bande passante réservée. De ce fait, certaines connexions sont susceptibles d'être bloquées même si le réseau est en mesure de les supporter. Néanmoins, le mode de réservation que nous avons choisi permet d'offrir des garanties strictes.

Aussi, nous n'avons pas étudié le cas de contraintes de QdS *soft*, ce qui rendrait le mécanisme de CAC moins conservateur. Pour ce faire, il faudrait considérer, pour chaque lien, des modèles de distributions de délai et de perte de paquets qui variraient en fonction de la charge du lien. Cette approche implique de tenir compte simultanément du routage et du partitionnement des paramètres de QdS et cela nécessiterait quelques modifications aux modèles proposés.

Par ailleurs, la topologie logique que nous avons choisie est basée sur les plus courts chemins. La construction de cette topologie ne tient pas compte du nombre de liens logiques traversant un lien physique, ce qui pourrait influencer le taux de blocage des connexions. Aussi, nous n'avons pas étudié l'impact de la granularité des connexions et du nombre de connexions sur les performances de nos algorithmes de contrôle d'admission. L'impact de la taille de la file d'attente sur nos résultats est un élément qu'il faudrait considérer car les calculs de délais et de perte de paquets en dépendent. Enfin,

nous n'avons pas non plus considéré une classification des connexions, ce qui mènerait à de la préemption lors de la réservation des ressources ou du traitement des paquets.

6.3 Travaux futurs

Les problèmes liés aux NGNs sont d'actualité. Par rapport à nos travaux, il serait premièrement intéressant de définir plus en détails les interfaces que nous allons utiliser dans l'architecture globale que nous avons proposée pour TISPAN. En effet, la définition des interfaces et des structures de données est un aspect important dans la mise en place d'une architecture. Toutefois, toute proposition dans ce sens devrait s'aligner avec les travaux du groupe TISPAN.

Un axe de recherche qui mériterait qu'on s'y attarde est la sécurité dans la nouvelle architecture. Premièrement, les communications entre les contrôleurs du réseau Ethernet et du réseau d'accès TISPAN doivent être sécurisées, ce qui pourrait se faire à l'aide d'un tunnel préconfiguré et de fonctions de cryptage. Ensuite, il faudrait aussi traiter plus en détails des fonctions de sécurité des nœuds frontières. En effet, ce sont ces nœuds qui jouent un rôle de protection du réseau contre toute intrusion extérieure.

Quant au contrôle d'admission de connexions, il faudrait considérer des intervalles pour le débit demandé par chaque connexion. En effet, au lieu de considérer une valeur rigide, chaque connexion pourrait définir une plage qui serait tolérable pour l'application transportée, ce qui permettrait de réduire le blocage. En outre, comme le délai et la perte de paquets sont liés d'une certaine manière, il faudrait explorer la voie consistant à définir une seule contrainte qui permettrait de satisfaire les deux critères de QdS pour une connexion donnée. Ce faisant, le nombre de contraintes à écrire dans un problème multi-contraintes serait réduit. Une autre avenue de recherche serait d'intégrer des contraintes *soft* de QdS pour obtenir un mécanisme de CAC moins conservateur.

Par ailleurs, la gigue est un paramètre qui mériterait aussi d'être pris en considération. Comme c'est une contrainte additive sur tout le chemin, une approche comme celle que nous avons proposée pour le problème de contrôle d'admission avec contraintes de délai pourrait être utilisée. Aussi, dans une implémentation réelle, le contrôleur pourrait

recevoir périodiquement les informations concernant la charge des différents liens et faire le contrôle d'admission en fonction des valeurs obtenues. Cela reviendrait à une approche de contrôle d'admission basée sur les mesures. Enfin, il serait intéressant d'implémenter un contrôleur utilisant nos modèles de CAC dans un logiciel comme OPNET afin d'en évaluer les performances de façon plus concrète.

BIBLIOGRAPHIE

3GPP, 3rd Generation Partnership Project, 2005, "IP Multimedia Subsystem (IMS), Stage 2, Release 7", Technical Specification Group Services and System Aspects, 3GPP TS 23.228 v7.1.0.

3GPP2, 3rd Generation Partnership Project 2, 2003, "All-IP Core Network Multimedia Domain: Overview", 3GPP2 X.S0013-000-0 v1.0.

IEEE, 2005, "Provider Bridge", http://www.ieee802.org/1/pages/802.1ad.html, Standard IEEE .

IEEE, 2004, "Traffic Class Expediting and Dynamic Multicast Filtering", publié dans 802.1D-2004, Annexe G, Standard IEEE.

IEEE, 2003, " Virtual Bridged Local Area Networks", Standard IEEE .

IEEE, 1996, " CSMA/CD Access Method and Physical Layer Specifications", Standard IEEE.

AGGARWAL R., 2004, "OAM mechanisms in MPLS layer 2 transport networks", IEEE Communications Magazine, vol. 42, No. 10, pp. 124-30.

AHUJA R.K., MAGNANTI T. L., ORLIN J.B., 1993, "Networks Flows", Prentice-Hall.

ALI M., CHIRUVOLU G., GE A., 2005a, "Traffic engineering in metro Ethernet", IEEE Network, vol. 19, No. 2, pp. 10-17.

ALI N.A., MOUFTAH H.T., GAZOR S., 2005b, "Online distributed statistical-delay MBAC with QoS guarantees for VPLS connections", ConTEL 2005, 8th International Conference on Telecommunications, 15-17 Juin 2005, Zagreb, Croatia: IEEE, vol. 2, pp. 383-90.

ANDERSSON L., PAPADIMITRIOU D., 2005, "Use of the Gnerealized Multi-Protocol Label Switching control plane for point-to-point Ethernet Label Switching". Internet Draft, draft-andersson-gels-bof-prep-01.txt, IETF.

ATIS, Alliance for Telecommunications Industry Solutions, 2004, "Part I: NGN definitions, Requirements and Architecture", ATIS Nest Generation Network (NGN) Framework.

ATRICA, 2003, "Migration from SONET/SDH to Carrier Ethernet In Metropolitan area", White Paper.

AWDUCHE D., BERGER L., Gan D., Li T., Srinivasan V., Swallow G., 2001, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, IETF.

AWDUCHE D., REKHTER Y., 2001, "Multiprotocol lambda switching: combining MPLS traffic engineering control with optical crossconnects", IEEE Communications Magazine, vol. 39, No. 3, pp. 111-16.

BADER A., WESTBERG L., KARAGIANNIS G., KAPPLER C., PHELAN T., 2006, " RMD-QOSM - The Resource Management in Diffserv QOS Model", Internet Draft, draft-ietf-nsis-rmd-06.txt, IETF.

BAKER F., ITURRALDE C., LE FAUCHEUR F., DAVIE B., 2001, "Aggregation of RSVP for IPv4 and IPv6 Reservations", RFC 3175, IETF.

BERGER L., 2003, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC3471, IETF.

BERGER L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, IETF.

BERGER L., GAN D., SWALLOW G., PAN P., TOMMASI F., MOLENDINI S., 2001, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, IETF.

BERNET Y., 2000, "The complementary roles of RSVP and differentiated services in the full-service QoS network", IEEE Communications Magazine, vol.38, No. 2, pp. 154-62.

BOTROFF P., 2005,"IEEE 802.1ah First Draft", IEEE 802.1 WG, Travail en cours.

BRADEN R., ZHANG L., BERSON S., HERZOG S., JAMIN S., 1997, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, IETF.

BRYANT S., PATE P., 2005, "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, IETF.

BURNS J.E., OTT T.J., KRZESINSKI A.E., MULLER K.E., 2003, "Path selection and bandwidth allocation in MPLS networks", Performance Evaluation, vol. 52, No. 2-3, pp. 133-52.

CAPONE A., FRATTA L., MARTIGNON F., 2003, "Dynamic Routing of Bandwidth Guaranteed Connections in MPLS Networks", International Journal on Wireless and Optical Communications, vol. 1, No. 1, pp. 75-86.

CHAMAS H., BJORKMAN W., ALI M.A., 2005, "A novel admission control scheme for Ethernet services", 2005 IEEE International Conference on Communications, 2005, ICC 2005, 16-20 May 2005, Seoul, Corée du Sud, vol.1, pp. 65-69.

CHENG K.T., LIN F.Y.S., 1995, "Minimax end-to-end delay routing and capacity assignment for virtual circuit networks", GLOBECOM'95 Global Telecommunications Conference, Nov. 95, Singapore, vol. 3, pp. 2134-2138.

CHIRUVOLU G., GE A., ELIE-DIT-COSAQUE D., ALI M., Rouyer J., 2004, " Issues and approaches on extending Ethernet beyond LANs", IEEE Communications Magazine, vol. 42, No. 3, pp. 80-6.

CHIUSSI F.A., KHOTIMSKY D.A., KRISHNAN S., 2002, "A network architecture for MPLS-based micro-mobility", 2002 IEEE Wireless Communications and Networking Conference Record. WCNC 2002, 17-21 March 2002. Orlando, FL, USA : IEEE, vol.2, pp. 549-55.

COLLE D., CHEYNS J., DEVELDER C., VAN BREUSEGEM E., ACKAERT A., PICKAVET M. et al. 2003, "GMPLS extensions for supporting advanced optical networking technologies", 2003 International Conference on Transparent Optical Networks - ICTON 2003, 29 June-3 July 2003, Warsaw, Poland : IEEE. vol.1, pp. 170-3.

CUI Y., XU Ke, WU J., 2003, "Precomputation for multiconstrained QoS routing in high-speed networks", INFOCOM 2003. Twenty-Second Annual Joint Conference

of the IEEE Computer and Communications Societies, 1-3 Avr. 2003, San Francisco, Arizona, USA: IEEE, vol. 2, pp. 1414-1424.

CUI Y., XU Ke, XU J., 2004, "Multiconstrained end-to-end admission control in core-stateless networks", Computer Science, vol. 3090, pp. 420-9.

DAVIS B., HUM S., DAVIS B., WALTON R., 2002, "Progress in last mile broadband wireless technologies", Proceedings of WIRELESS 2002, 8-10 July 2002, Calgary, Alta., Canada : TRLabs, vol.1, pp. 356-73.

DIAS R.A., CAMPONOGARA E., FARINES J.-M., WILLRICH R., CAMPESTRINI A., 2003, "Implementing traffic engineering in MPLS-based IP networks with Lagrangean relaxation", Eighth IEEE International Symposium on Computers and Communication, 30 Juin – 3 Juil. 2003, Kemer-Antalya, Turquie : IEEE, vol. 1, pp. 373-378.

DIXIT S., 2003, "IP over WDM building the next generation optical internet", Hoboken N.J., Wiley-Interscience.

ELANGOVAN A., 2005, "Efficient multicasting and broadcasting in layer 2 provider backbone networks ", IEEE Communications Magazine, vol. 43, No. 11, pp. 166-70.

FEDYK D., ALLAN D., 2006, "GMPLS control of Ethernet IVL Switches". Internet Draft, draft-fedyk-gmpls-ethernet-ivl-00.txt, IETF.

FEUZEU T., COUSIN B., 2005, " A new Scheme for Interconnecting LANs with Label Switching Bridges", IEEE Conference on Local Computer Networks 30th Anniversary, LCN'05, 15-17 Nov. 2005, Sydney, Australia : IEEE, pp. 303-10

GENG-SHENG K., QING H., 2003, "GMPLS-based micro-mobility for next-generation mobile broadband IP networks", ICCT 2003 - International Conference on Communication Technology, 9-11 April 2003, Beijing, China : Beijing Univ, Posts & Telecommun. Press. vol.1, pp. 32-7.

GHANWANI A., PACE W., SRINIVASAN V., SMITH A., SEAMAN M., 2000, " A Framework for Integrated Services Over Shared and Switched IEEE 802 LAN Technologies", RFC2816, IETF.

HUNG-SHIH C., KUOCHEN W., 2003, "An all-MPLS approach for UMTS 3G core networks", 2003 IEEE 58th Vehicular Technology Conference, VTC 2003-Fall, 6-9 Oct. 2003, Orlando, FL, USA : IEEE, vol.4, pp. 2338-42.

ITU-T, International Telecommunication Union, 2004a, "Next Generation Networks-Frameworks and functional architecture models", Y.2001.

ITU-T, International Telecommunication Union, 2004b, "General overview of NGN", Next Generation Networks-Frameworks and functional architecture models, "General principles and general reference model for Next Generation", Y.2011.

ITU-T, International Telecommunication Union, 2005a, "Interfonctionnement des des réseaux Ethernet et MPLS – Interfonctionnement dans le plan utilisateur ", Y.1415.

ITU-T, International Telecommunication Union, 2005b, "Revision 7 of TR-123.qos ", TR-123.qos.

ITU-T, International Telecommunication Union, 2006, "A revised Draft Text for TR-enet", FGNGN-OD-00202, TR-enet.

JAFFE J. M., 1984, “Algorithms for Finding Paths with Multiple Constraints”, Networks, vol. 14, 95–116.

JAIHYUNG C., 2005, “Label switched ethernet technology”, 7th International Conference on Advanced Communication Technology, ICACT, 21-23 Fev. 2005, Paris, France. vol.1, pp. 619-23.

JAIHYUNG C., 2006, “Label Switching Ethernet (LSE) Architecture”. Internet Draft, draft-jaihyumg-lse-architecture-00.txt, IETF.

JONG-MOON C., 2001, "Wireless multiprotocol label switching (WMPLS)",Conference Record, Thirty-Fifth Asilomar Conference on Signals, Systems and Computers, 4-7 Nov. 2001, Pacific Grove, CA, USA : IEEE, vol.1, pp. 679-83.

KELLIHER, J.C., AGHVAMI H.A., 2004. "The 4th generation conjointed network 'all-optical' path routed communication networks: foundation for a 4G broadband mobile network", Telecommunications Quality of Service: The Business of Success (QoS 2004), 2-3 March 2004, London, UK : IEE, pp. 32-6.

KHAN S., LI K.F., MANNING E.G., WATSON R., SHOJA G.C., 2003, "Optimal quality of service routing and admission control using the utility model", Future Generation Computer Systems, vol. 19, No. 7, pp. 1063-73.

KODIALAM M. S., LAKSHMAN T. V., 2000, "Minimum interference routing with applications to MPLS traffic engineering", Proceedings of INFOCOM (2), 26-30 Mars 2000, Tel-Aviv, Israel: IEEE, pp. 884-893.

KUIPERS F., VAN MIEGHEM P., KORKMAZ T., KRUNZ M., 2002, "An overview of constraint-based path selection algorithms for QoS routing", IEEE Communications Magazine, vol. 40, No. 12, pp. 50-5.

LUCIDARME T., 2002, "Principes de radiocommunication de troisième génération GSM, GPRS, UMTS", Paris : Vuibert Informatique.

MAHMOODIAN A., HARING G., 2000,. "Mobile RSVP with dynamic resource sharing", Proceedings of IEEE Conference on Wireless Communications and Networking, 23-28 Sept. 2000, Chicago, IL, USA : IEEE, vol.2, pp. 896-901.

MANNER J., FU, X., 2004, "Analysis of Existing Quality of Service Signaling Protocols", Internet Draft, draft-ietf-nsis-signalling-analysis-05.txt, IETF.

MARTINI L., HERON G., 2005, "Encapsulation Methods for Transport of Ethernet Over MPLS Networks". Internet Draft, draft-ietf-pwe3-ethernet-encap-11.txt, IETF.

MARTINI L., NASSER E., VOGELSANG S., TAPPAN D., SHIRRON J., ROSEN E., smith T., HAMILTON A., 2006,"Transport of Layer 2 Frames Over MPLS". Internet Draft, draft-martini-l2circuit-trans-mpls-17.txt, IETF.

MATHY L., HUTCHISON D., SCHMID S., Coulson G., 1999, "Improving RSVP for better support of Internet multimedia communications ", Proceedings of ICMCS99: IEEE Multimedia Systems '99: International Conference on Multimedia Computing and Systems, 7-11 June 1999, Florence, Italy : IEEE Comput. Soc. vol.2, pp. 102-6.

MEF, Metro Ethernet Forum, 2004a, "Ethernet Services definitions-Phase I", Technical Specification MEF 6.

MEF, Metro Ethernet Forum, 2004b, "EMS-NMS Information Model", Technical Specification MEF 7.

MEF, Metro Ethernet Forum, 2005, "Requirements for Management of Metro Ethernet Phase 1 Network Elements", Technical Specification MEF 15.

Misra I.S., Banerjee A., 2003, "MPLS based mobility framework in 4G architectures", IEEE TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region, 15-17 Oct. 2003, Bangalore, India : Allied Publishers Pvt. Ltd. vol.2, pp. 670-4.

MOON B., Aghvami H., 2001, "RSVP extensions for real-time services in wireless mobile networks", IEEE Communications Magazine, vol. 39, No. 12, pp. 52-9.

MSF, Multiservice Switching Forum, 2005a, "MSF Release 2 Architecture", MSF-ARCH-002.0-FINAL.

MSF, Multiservice Switching Forum, 2005b, "MSF Release 2 Implementation Guidelines".

MSF, Multiservice Switching Forum, 2005c, "Bandwidth Management in Next generation Packet Networks ", MSF Technical Report, MSF-TR-ARCH-005-FINAL.

NORDSTROM E., DZIONG Z., 2006, "CAC and routing for multi-service networks with blocked wide-band calls delayed, part I: exact link MDP framework", European Transactions on Telecommunications, vol.17, No. 1, pp. 21-36.

NSP, 2003, "The Business Case for an Optical Ethernet Metro Area Network",

http://www.atrica.com/body/products/whitepapers/Public_ROI_Study.pdf, Network Strategy Partner.

OULAI D., CHAMBERLAND S., PIERRE S., 2006a, "A New Routing-Based Admission Control for MPLS Networks", IEEE Communications Letters, vol. 11, No. 2, pp. 216-8.

OULAI D., CHAMBERLAND S., PIERRE S., 2006b, "Routing and Admission Control with Multiconstrained End-to-End Quality of Service in MPLS Networks", Soumis à Computer Communications.

OULAI D., CHAMBERLAND S., PIERRE S., 2006c, "End-to-End Quality of Service Constrained Routing and Admission Control for MPLS Networks", Soumis à Computer Networks.

OULAI D., CHAMBERLAND S., PIERRE S., 2006e, "End-to-End Packet Loss Constrained Routing and Admission Control for MPLS Networks", Accepté à la conference CCECE07.

PACKETCABLE, 2005a, "Architecture Framework Technical Report", PacketCable 1.5, PKT-TR-ARCH1.V01-050128.

PACKETCABLE, 2005b, "Multimedia Specification", PacketCable Specification, PKT-SP-MM-I03-051221.

PAPADIMITRIOU D., DOTARO E., VIGOUREUX M., 2005, "Ethernet layer 2 label switched paths (LSP)", Next Generation Internet Networks, 2005 18-20 April 2005, Rome, Italy. pp. 188-94.

PAREKH A.K., GALLAGER R.G., 1994, "A generalized processor sharing approach to flow control in integrated services networks: the multiple node case", IEEE/ACM Transactions on Networking, vol.2, No. 2, pp. 137-50.

PASKALIS S., KALOXYLOS A., ZERVAS E., MERAKOS L., 2003, "An efficient RSVP-Mobile IP interworking scheme", Mobile Networks and Applications, vol. 8, No. 3, pp. 197-207.

PING P., SCHULZRINNE H., 1997, "Staged refresh timers for RSVP", GLOBECOM 97, IEEE Global Telecommunications Conference, Conference Record, 3-8 Nov. 1997, Phoenix, AZ, USA : IEEE, vol.3, pp. 1909-13.

PROJET AGAVE, 2004, "Architecture GMPLS : Analyse, Validation, Expériementation ", Rapport d'étude, École Nationale Supérieure des Télécommunications de Bretagne.

RAMJEE R., LA PORTA T.F., SALGARELLI L., THUEL S., VARADHAN K., LI L., 2000, "IP-based access network infrastructure for next-generation wireless data networks", IEEE Personal Communications, vol. 7, No. 4, pp. 34-41.

SEAMAN M., SMITH A., CRAWLEY E., WROCLAWSKI J., 2000, " Integrated Service Mappings on IEEE 802 Networks", RFC2815, IETF.

SHOU-CHIHLO G ., WEN-TSUEN C., JEN-CHI L., 2004, "Architecture for mobility and QoS support in all-IP wireless networks", IEEE Journal on Selected Areas in Communications, vol. 22, No. 4, pp. 691-705.

SPITLER S.L., LEE D.C, 2003, "Optimal call admission control under packet and call level QoS constraints and effect of buffering", International Journal of

Communication Systems, vol. 16, No. 7, pp. 647-62.

TAHA A.E.M., HASSANEIN H., Mouftah H.T., 2004, "Integrated solutions for wireless MPLS and Mobile IP: current status and future directions", Canadian Conference on Electrical and Computer Engineering 2004, 2-5 May 2004, Niagara Falls, Ont., Canada : IEEE, vol.3, pp. 1463-6.

TISPAN, Telecommunications and Internet converged Services and Protocols for Advanced Networking, ETSI, 2005a, "NGN Functional Architecture Release 1", ETSI draft ES 282001v1.1.1

TISPAN, Telecommunications and Internet converged Services and Protocols for Advanced Networking, ETSI, 2005b, " Resource and Admission Control Subsystem (RACS)", ETSI draft ES 282003v1.6.5.

TISPAN, Telecommunications and Internet converged Services and Protocols for Advanced Networking, ETSI, 2005c," Tispan_NGN; Release 1: Release Definition ", ETSI draft ES TR 00001v0.4.2.

TSCHOFENIG H., GRAVEMAN R., 2004, "RSVP Security Properties", Internet Draft, draft-ietf-nsis-rsvp-sec-properties-05,txt, IETF.

UIT-T, Y.1281, 2003 "Services mobiles sur MPLS".

VOGT C., 2002, "Admission control and resource reservation on the internet", ACM SIGSOFT Software Engineering Notes, vol. 27, No. 3, pp. 80-7.

WESTBERG L., CSASZAR A., KARAGIANNIS G., MARQUETANT A., PARTAIN

D., Pop O., Rexhepi V., Szabo R., Takacs A., 2002., "Resource management in DiffServ (RMD): a functionality and performance behavior overview", PfHSN 2002 - . 7th IFIP/IEEE International Workshop on Protocols for High Speed Networks , 22-24 April 2002, Berlin, Germany, Lecture Notes in Computer Science vol.2334, pp. 17-34.

WESTBERG L., BADER A., PARTAIN D., REXHEPI V., 2003., "A Proposal for RSVPv2-NSLP", Internet Draft, draft-westberg-proposal-for-rsvpv2-nslp-00,txt, IETF.

WIDYONO R., 1994, "The Design and Evaluation of Routing Algorithms for Real-Time Channels", Univ. of California at Berkeley, Tech. Rep., ICSI TR-94-024.

WROCLAWSKI J., 1997, "The Use of RSVP with IETF Integrated Services", RFC 2210, IETF.

WU Y., HUI J., SUN, H., 2003, "Fast restoring gigabit wireless networks using a directional mesh architecture", Computer Communications, vol. 26, No. 17, pp. 1957-64.

YAVATKAR R., HOFFMAN D., BERNET Y., BAKER F., SPEER M., 2000, " SBM (Subnet Bandwidth Manager):A Protocol for RSVP-based Admission Control over IEEE 802-style networks", RFC2814, IETF.

YI Han Z., MAKRAKIS D., PRIMAK S., YUN BO H., 2002, "Dynamic support of service differentiation in wireless networks ", IEEE CCECE2002. Canadian Conference on Electrical and Computer Engineering. Conference Proceedings, 12-15 May 2002. Winnipeg, Man., Canada : IEEE, vol.3, pp. 1325-30.

YILE G, ANTONIOU Z., DIXIT S., 2002, "IP transport in 3G radio access networks: an MPLS-based approach", 2002 IEEE Wireless Communications and Networking Conference Record. WCNC 2002, 17-21 March 2002, Orlando, FL, USA : IEEE, vol.1, pp. 11-17.

YUAN X., 2002, "Heuristic algorithms for multiconstrained quality-of-service routing", IEEE/ACM Transactions on Networking, vol. 10, No. 2, pp. 244-256.

YUGE T., TSUBOUCHI K., 2002, "Realization of high speed all IP network system", Record of Electrical and Communication Engineering Conversazione Tohoku University, vol. 71, No.1, pp. 169-72.

ZENG H., DOU J., XU D., 2005, "Replace MPLS with EPFTS to build a SUPANET", 2005 Workshop on High Performance Switching and Routing,, HPSR , 12 -14 May 2005, Hong-Kong, Chine. pp. 39-43.

ZHANG L., DEERING S., ESTRIN D., SHENKER S., ZAPPALA D., 1993, " RSVP: a new resource ReSerVation Protocol ", IEEE Network, vol. 7, No. 5, pp. 8-18.

ZHONG R., CHEN-KHONG T., CHUN-CHOONG F., CHI-CHUNG K., 2001, "Integration of mobile IP and multi-protocol label switching", Proceedings of International Conference on Communications, 11-14 June 2001, Helsinki, Finland : IEEE, vol.7, pp. 2123-7.

ANNEXE

ANNEXE A : Automates pour le réseau TISPAN

Automates pour la réservation de ressources

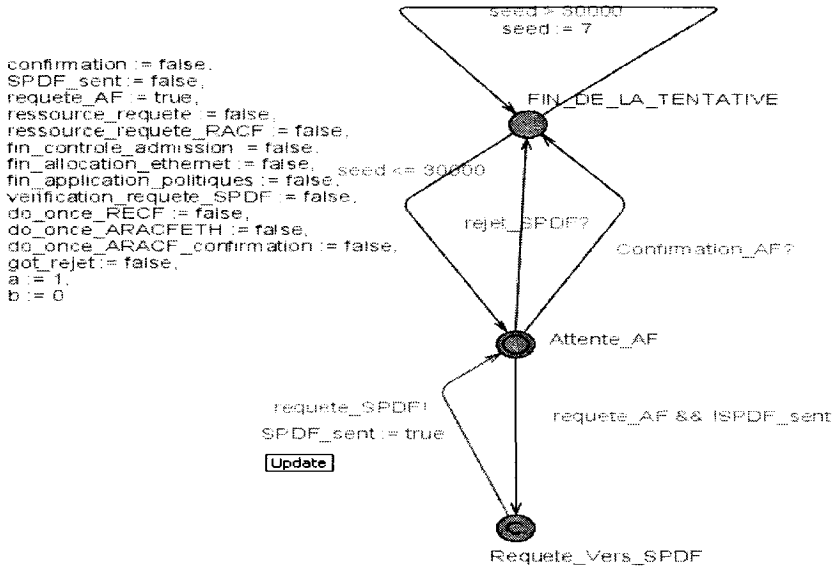

Figure A.1 Automate AF pour la réservation de ressources

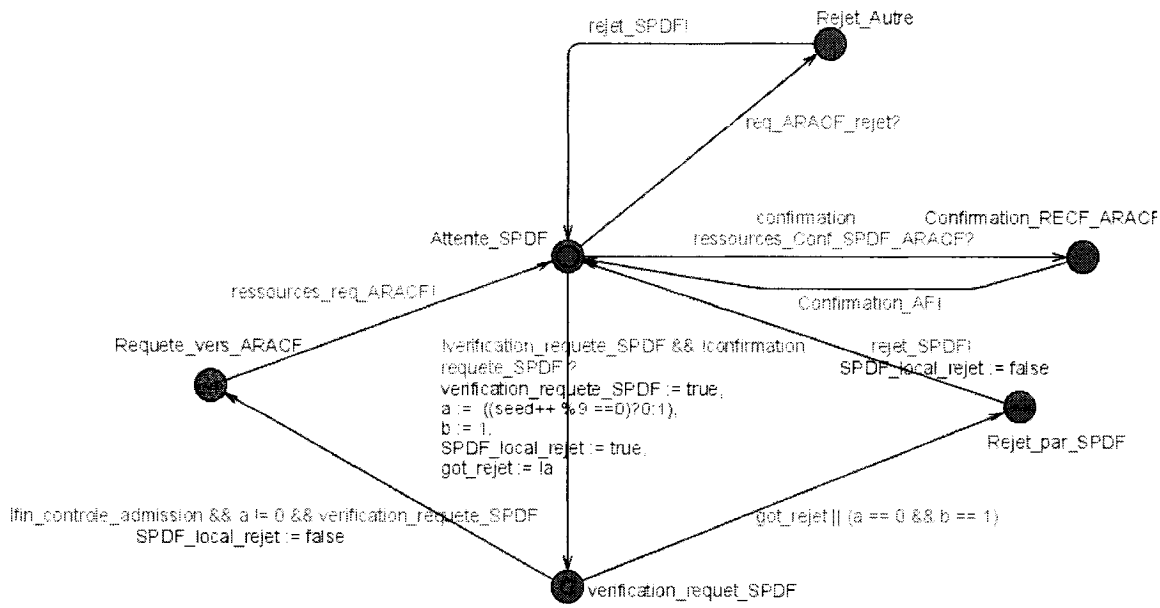

Figure A.2 Automate SPDF pour la réservation de ressources

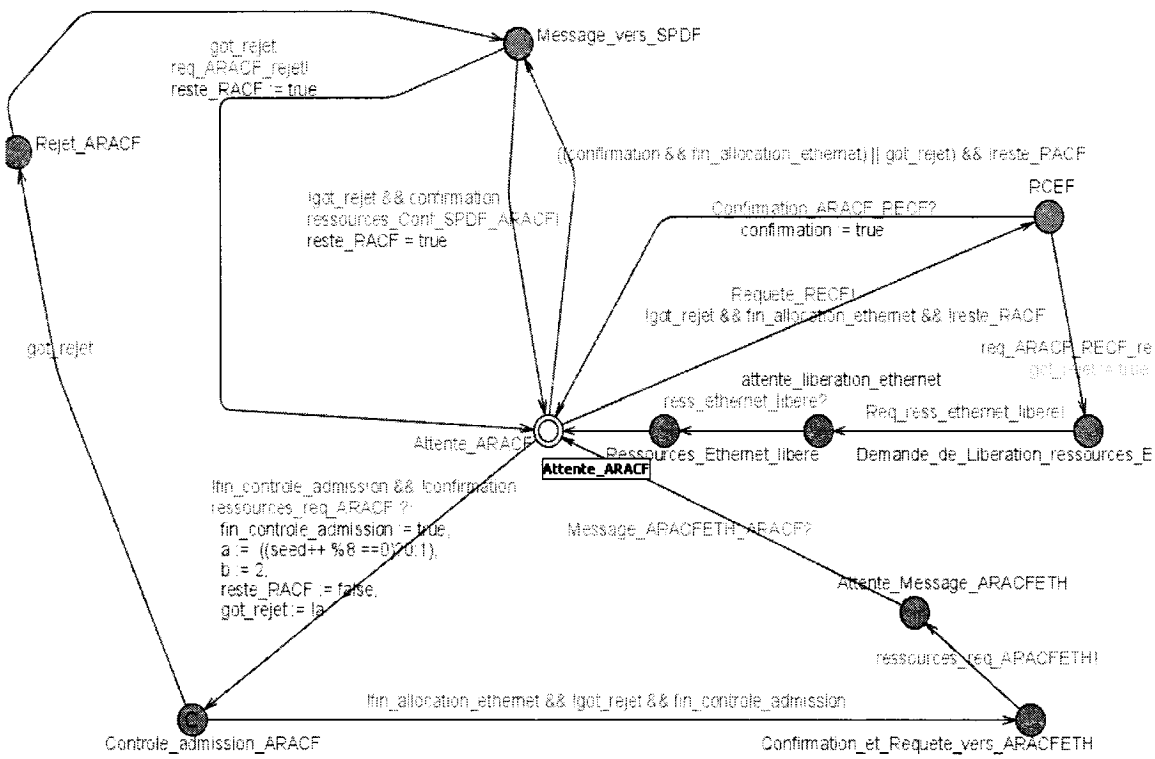

Figure A.3 Automate A-RACF pour la réservation de ressources

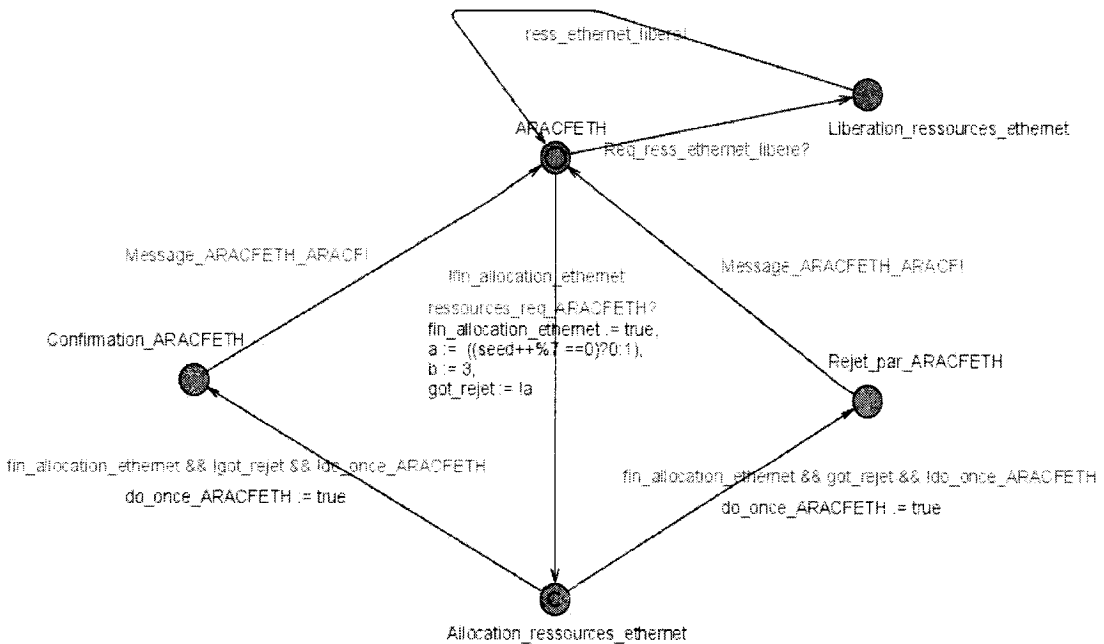

Figure A.4 Automate A-RACFETH pour la réservation de ressources

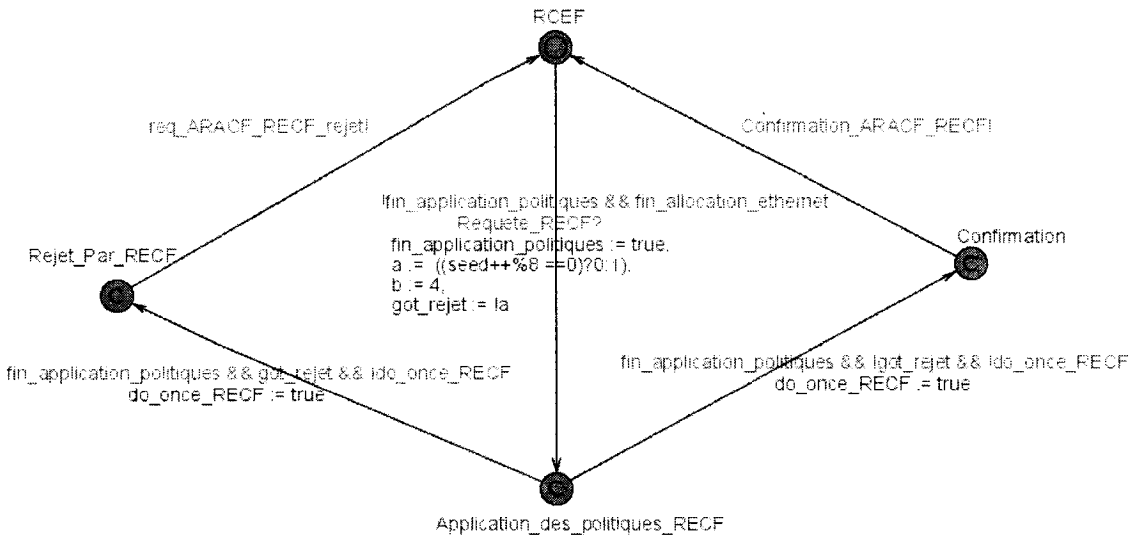

Figure A.5 Automate RCEF pour la réservation de ressources

Automates pour la libération de ressources

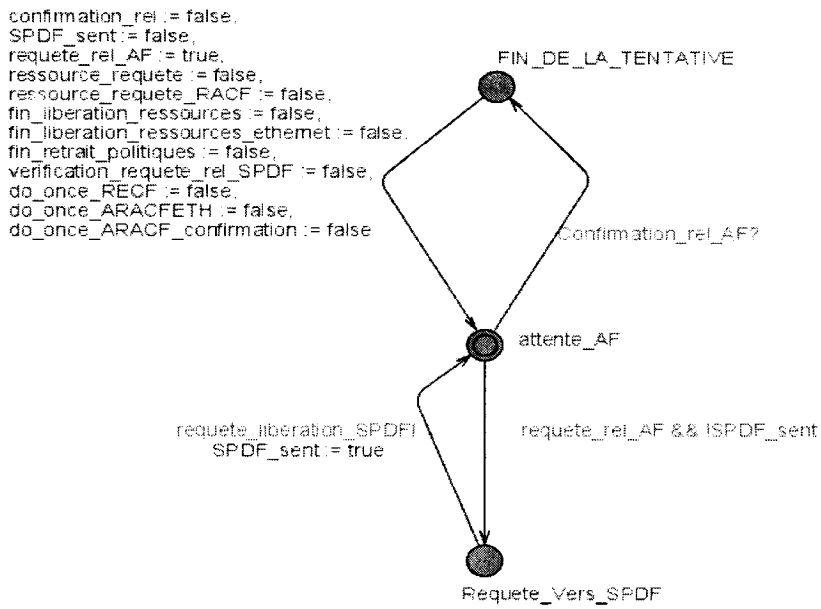

Figure A.6 Automate AF pour la libération de ressources

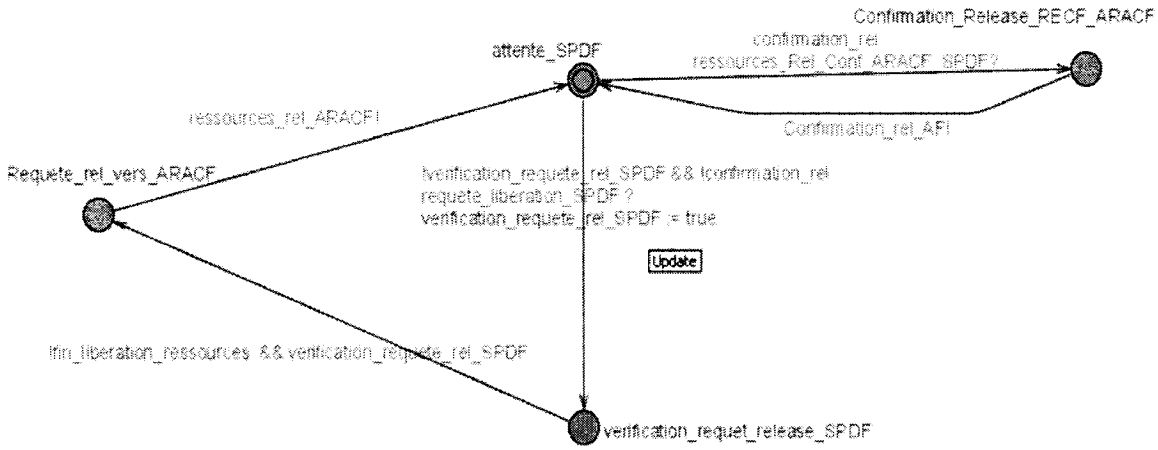

Figure A.7 Automate SPDF pour la libération de ressources

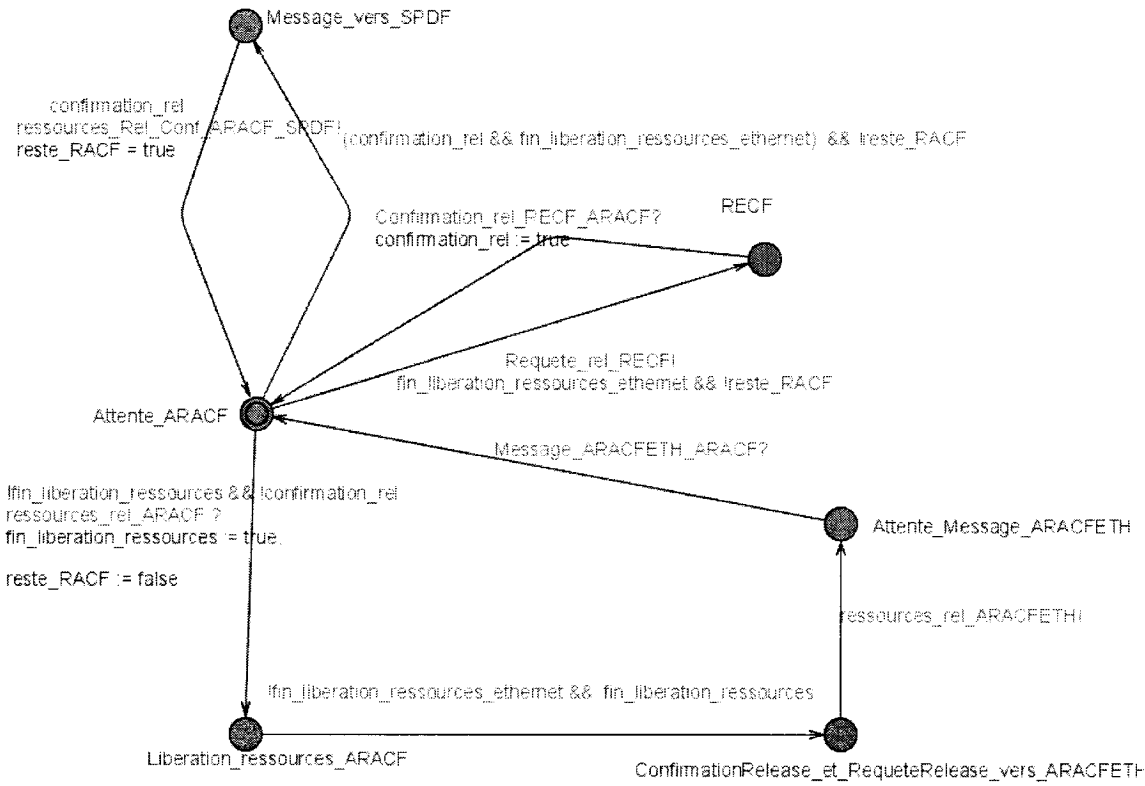

Figure A.8 Automate A-RACF pour la libération de ressources

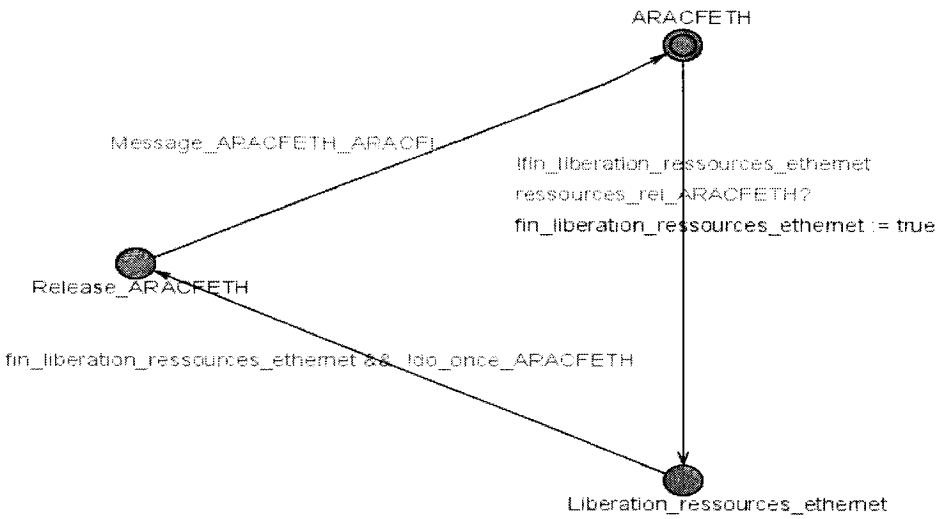

Figure A.9 Automate A-RACFETH pour la libération de ressources

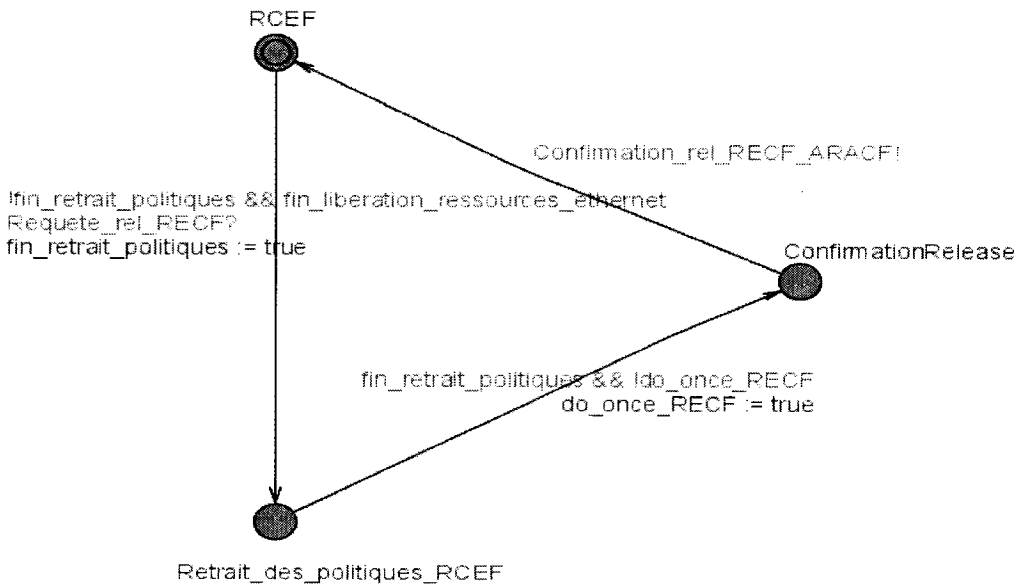

Figure A.10 Automate RCEF pour la libération de ressources